

# SN32F260 系列

## 用户参考手册

SN32F268/267/265/264/2641/263

# SONiX 32 位 Cortex-M0 单片机

SONiX 公司保留对以下所有产品在可靠性，功能和设计方面的改进作进一步说明的权利。SONiX 不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，SONiX 的产品不是专门设计来应用于外科植入、生命维持和任何 SONiX 产品的故障会对个体造成伤害甚至死亡的领域。如果将 SONiX 的产品应用于上述领域，即使这些是由 SONiX 在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接产生的律师费用，并且用户保证 SONiX 及其雇员、子公司、分支机构和销售商与上述事宜无关。

## 修改记录

版本	时间	修改说明
V1.2	2017/03	中文版初版。

## 目录

修改记录.....	2
目录.....	3
1 产品简介.....	8
1.1 功能特性.....	8
1.2 系统框图.....	9
1.3 时钟产生框图.....	10
1.4 引脚配置.....	11
1.5 引脚说明.....	15
1.6 引脚电路结构图.....	16
2 中央处理器（CPU）.....	17
2.1 存储器.....	17
2.2 系统TICK定时器.....	18
2.2.1 操作.....	18
2.2.2 SYSTICK用法说明及技巧.....	18
2.2.3 SYSTICK寄存器.....	19
2.2.3.1 系统Tick定时器控制和状态寄存器（SYSTICK_CTRL）.....	19
2.2.3.2 系统Tick定时器重装值寄存器（SYSTICK_LOAD）.....	19
2.2.3.3 系统Tick定时器当前值寄存器（SYSTICK_VAL）.....	19
2.2.3.4 系统Tick定时器校准值寄存器（SYSTICK_CALIB）.....	19
2.3 嵌套向量中断控制器（NVIC）.....	20
2.3.1 中断和异常向量.....	20
2.3.2 NVIC寄存器.....	21
2.3.2.1 IRQ0~31 中断设置使能寄存器（NVIC_ISER）.....	21
2.3.2.2 IRQ0~31 中断清零使能寄存器（NVIC_ICER）.....	21
2.3.2.3 IRQ0~31 中断挂起设置寄存器（NVIC_ISPR）.....	21
2.3.2.4 IRQ0~31 中断挂起清零寄存器（NVIC_ICPR）.....	21
2.3.2.5 IRQ0~31 中断优先级寄存器（NVIC_IPRn）（n=0~7）.....	22
2.4 应用中断和复位控制（AIRC）.....	23
2.5 编译选项列表（CODE OPTION）.....	23
2.6 核心寄存器简介.....	24
3 系统控制.....	25
3.1 复位.....	25
3.1.1 上电复位（POR）.....	25
3.1.2 看门狗复位（WDT RESET）.....	25
3.1.3 掉电复位.....	26
3.1.3.1 掉电复位简述.....	26
3.1.3.2 系统工作电压.....	26
3.1.3.3 掉电复位性能改进.....	27
3.1.4 外部复位.....	28
3.1.4.1 RC复位电路.....	28
3.1.4.2 二极管&RC复位电路.....	29
3.1.4.3 稳压二极管复位电路.....	29
3.1.4.4 电压偏置复位电路.....	30
3.1.4.5 外部IC复位电路.....	30
3.1.5 软件复位.....	31
3.2 系统时钟.....	32
3.2.1 内部RC时钟源.....	32
3.2.1.1 内部高速RC振荡器（IHRC）.....	32
3.2.1.2 内部低速RC振荡器（ILRC）.....	32
3.2.2 系统时钟（SYSCLK）选择.....	33
3.2.3 CLOCK-OUT功能.....	33
3.3 系统控制寄存器 0.....	34
3.3.1 模拟模块控制寄存器（SYS0_ANBCTRL）.....	34
3.3.2 时钟源状态寄存器（SYS0_CSST）.....	34
3.3.3 系统时钟配置寄存器（SYS0_CLKCFG）.....	34
3.3.4 AHB时钟预分频寄存器（SYS0_AHBCP）.....	34
3.3.5 系统复位状态寄存器（SYS0_RSTST）.....	35
3.3.6 LVD控制寄存器（SYS0_LVDCTRL）.....	35
3.3.7 外部复位引脚控制寄存器（SYS0_EXRSTCTRL）.....	35
3.3.8 SWD引脚控制寄存器（SYS0_SWDCTRL）.....	36

3.3.9	中断向量映射寄存器 (SYS0_IVTM)	36
3.3.10	噪音检测控制寄存器 (SYS0_NDTCTRL)	36
3.3.11	噪音检测状态寄存器 (SYS0_NDTSTS)	36
3.3.12	Anti-EFT Ability控制寄存器 (SYS0_ANTIEFT)	36
3.4	系统控制寄存器 1	37
3.4.1	AHB时钟使能寄存器 (SYS1_AHBCLKEN)	37
3.4.2	APB时钟预分频寄存器 1 (SYS1_APBPCP1)	38
4	系统操作模式	39
4.1	概述	39
4.2	普通模式	39
4.3	低功耗模式	40
4.3.1	睡眠模式	40
4.3.2	深度睡眠模式	40
4.4	系统唤醒	41
4.4.1	概述	41
4.4.2	唤醒时间	41
4.5	PMU状态结构图	42
4.6	操作模式Comparsion表	42
4.7	PMU寄存器	43
4.7.1	电源控制寄存器 (PMU_CTRL)	43
5	GPIO端口	44
5.1	概述	44
5.2	GPIO模式	44
5.3	GPIO寄存器	45
5.3.1	GPIO Pn数据寄存器 (GPIOn_DATA) (n=0,1,2,3)	45
5.3.2	GPIO Pn模式寄存器 (GPIOn_MODE) (n=0,1,2,3)	45
5.3.3	GPIO Pn配置寄存器 (GPIOn_CFG) (n=0,1,2,3)	45
5.3.4	GPIO Pn中断SENSE寄存器 (GPIOn_IS) (n=0,1,2,3)	47
5.3.5	GPIO Pn中断双边沿SENSE寄存器 (GPIOn_IBS) (n=0,1,2,3)	47
5.3.6	GPIO Pn中断事件寄存器 (GPIOn_IEV) (n=0,1,2,3)	47
5.3.7	GPIO Pn中断使能寄存器 (GPIOn_IE) (n=0,1,2,3)	47
5.3.8	GPIO Pn中断源状态寄存器 (GPIOn_RIS) (n=0,1,2,3)	47
5.3.9	GPIO Pn中断清零寄存器 (GPIOn_IC) (n=0,1,2,3)	48
5.3.10	GPIO Pn位设置操作寄存器 (GPIOn_BSET) (n=0,1,2,3)	48
5.3.11	GPIO Pn位清零操作寄存器 (GPIOn_BCLR) (n=0,1,2,3)	48
6	16位带捕获功能定时器	49
6.1	概述	49
6.2	特性	49
6.3	引脚说明	49
6.4	框图	50
6.5	定时器操作	51
6.5.1	边沿对齐向上计数模式	51
6.6	PWM	52
6.6.1	PWM模式 1	52
6.6.2	PWM模式 2	53
6.7	CT16Bn寄存器	54
6.7.1	CT16Bn定时器控制寄存器 (CT16Bn_TMRCTRL) (n=0,1)	54
6.7.2	CT16Bn定时器计数器寄存器 (CT16Bn_TC) (n=0,1)	54
6.7.3	CT16Bn前置分频寄存器 (CT16Bn_PRE) (n=0,1)	54
6.7.4	CT16Bn前置分频计数器寄存器 (CT16Bn_PC) (n=0,1,2)	54
6.7.5	CT16Bn计数控制寄存器 (CT16Bn_CNTCTRL) (n=0)	55
6.7.6	CT16Bn匹配控制寄存器 (CT16Bn_MCTRL) (n=0)	55
6.7.7	CT16Bn匹配控制寄存器 (CT16Bn_MCTRL) (n=1)	56
6.7.8	CT16Bn匹配控制寄存器 2 (CT16Bn_MCTRL2) (n=1)	58
6.7.9	CT16Bn匹配控制寄存器 3 (CT16Bn_MCTRL3) (n=1)	60
6.7.10	CT16Bn匹配寄存器 0 (CT16Bn_MR) (n=0)	60
6.7.11	CT16Bn匹配寄存器 0~19,21~23 (CT16Bn_MR0~19,21~23) (n=1)	61
6.7.12	CT16Bn捕获控制寄存器 (CT16Bn_CAPCTRL) (n=0)	61
6.7.13	CT16Bn捕获 0 寄存器 (CT16Bn_CAP0) (n=0)	61
6.7.14	CT16Bn外部匹配寄存器 (CT16Bn_EM) (n=1)	62
6.7.15	CT16Bn外部匹配控制寄存器 (CT16Bn EMC) (n=1)	63
6.7.16	CT16Bn外部匹配控制寄存器 2 (CT16Bn EMC2) (n=1)	65
6.7.17	CT16Bn PWM控制寄存器 (CT16Bn_PWMCTRL) (n=1)	66
6.7.18	CT16Bn PWM控制寄存器 2 (CT16Bn_PWMCTRL2) (n=1)	68

6.7.19	CT16Bn PWM使能寄存器 (CT16Bn_PWMENB) (n=1)	69
6.7.20	PWM IO使能寄存器 (CT16Bn_PWMIOENB) (n=1)	71
6.7.21	CT16Bn定时器中断源状态寄存器 (CT16Bn_RIS) (n=0,1)	73
6.7.22	CT16Bn定时器中断清零寄存器 (CT16Bn_IC) (n=0,1)	75
7	看门狗定时器 (WDT)	76
7.1	概述	76
7.2	框图	77
7.3	WDT寄存器	78
7.3.1	看门狗配置寄存器 (WDT_CFG)	78
7.3.2	看门狗定时器常量寄存器 (WDT_TC)	78
7.3.3	看门狗Feed寄存器 (WDT_FEED)	78
8	SPI	79
8.1	概述	79
8.2	特性	79
8.3	引脚说明	79
8.4	接口说明	80
8.4.1	SPI	80
8.4.2	通讯流程	81
8.4.2.1	单帧	81
8.4.2.2	多帧	81
8.5	AUTO-SEL	82
8.6	SPI寄存器	83
8.6.1	SPI n控制寄存器 0 (SPIn_CTRL0) (n=0)	83
8.6.2	SPI n控制寄存器 1 (SPIn_CTRL1) (n=0)	84
8.6.3	SPI n时钟分频寄存器 (SPIn_CLKDIV) (n=0)	84
8.6.4	SPI n状态寄存器 (SPIn_STAT) (n=0)	84
8.6.5	SPI n中断使能寄存器 (SPIn_IE) (n=0)	85
8.6.6	SPI n中断源状态寄存器 (SPIn_RIS) (n=0)	85
8.6.7	SPI n中断清零寄存器 (SPIn_IC) (n=0)	85
8.6.8	SPI n数据寄存器 (SPIn_DATA) (n=0)	86
8.6.9	SPI n数据Fetch寄存器 (SPIn_DF) (n=0)	86
9	I2C	87
9.1	概述	87
9.2	特性	87
9.3	引脚说明	88
9.4	特性波形图	88
9.5	I2C主机模式	89
9.5.1	主机发送模式	89
9.5.2	主机接收模式	89
9.5.3	仲裁	89
9.6	I2C从机模式	90
9.6.1	从机发送模式	90
9.6.2	从机接收模式	90
9.7	I2C寄存器	91
9.7.1	I2C n控制寄存器 (I2Cn_CTRL) (n=0)	91
9.7.2	I2C n状态寄存器 (I2Cn_STAT) (n=0)	92
9.7.3	I2C n TX数据寄存器 (I2Cn_TXDATA) (n=0)	93
9.7.4	I2C n RX数据寄存器 (I2Cn_RXDATA) (n=0)	93
9.7.5	I2C n从机地址 0 寄存器 (I2Cn_SLVADDR0) (n=0)	93
9.7.6	I2C n从机地址 1~3 寄存器 (I2Cn_SLVADDR1~3) (n=0)	93
9.7.7	I2C n SCL高电平时间寄存器 (I2Cn_SCLHT) (n=0)	94
9.7.8	I2C n SCL低电平时间寄存器 (I2Cn_SCLLT) (n=0)	94
9.7.9	I2C n超时控制寄存器 (I2Cn_TOCTRL) (n=0)	94
10	USB FS设备接口	95
10.1	概述	95
10.2	特性	95
10.3	引脚说明	95
10.4	框图	96
10.5	USB SRAM访问	96
10.6	USB机构	97
10.7	USB中断	97
10.8	USB枚举	97
10.9	USB寄存器	98
10.9.1	USB中断使能寄存器 (USB_INTEN)	98

10.9.2	USB中断事件状态寄存器 (USB_INSTS)	99
10.9.3	USB中断事件状态清零寄存器 (USB_INSTSC)	100
10.9.4	USB设备地址寄存器 (USB_ADDR)	101
10.9.5	USB配置寄存器 (USB_CFG)	101
10.9.6	USB信号控制寄存器 (USB_SGCTL)	101
10.9.7	USB EP0 控制寄存器 (USB_EP0CTL)	102
10.9.8	USB Epn控制寄存器 (USB_EPnCTL, n = 1 ~ 4)	102
10.9.9	USB Epn数据Toggle寄存器 (USB_EPTOGGLE)	103
10.9.10	USB Epn缓存器偏移寄存器 (USB_EPnBUFOS, n = 1 ~ 4)	103
10.9.11	USB帧数字寄存器 (USB_FRMNO)	103
10.9.12	USB PHY参数寄存器 (USB_PHYPRM)	103
10.9.13	USB PHY参数寄存器 2 (USB_PHYPRM2)	104
10.9.14	USB PHY参数寄存器 (USB_PS2CTL)	104
10.9.15	USB R/W地址寄存器 (USB_RWADDR)	104
10.9.16	USB R/W数据寄存器 (USB_RWDATA)	104
10.9.17	USB R/W状态寄存器 (USB_RWSTATUS)	104
10.9.18	USB R/W地址寄存器 2 (USB_RWADDR2)	105
10.9.19	USB R/W数据寄存器 2 (USB_RWDATA2)	105
10.9.20	USB R/W状态寄存器 2 (USB_RWSTATUS2)	105
11	FLASH	106
11.1	概述	106
11.2	嵌入式FLASH存储器	106
11.3	特性	106
11.4	机构	107
11.5	读操作	107
11.6	编程/擦除	107
11.7	嵌入式引导加载程序	107
11.8	FLASH存储控制器 (FMC)	108
11.8.1	代码加密 (CS)	108
11.8.2	编程FLASH存储器	109
11.8.3	擦除操作	109
11.8.3.1	按页擦除	109
11.8.3.2	批量擦除	109
11.9	读保护	109
11.10	HW CHECKSUM	109
11.11	FMC寄存器	110
11.11.1	Flash低电压控制寄存器 (FLASH_LPCTRL)	110
11.11.2	Flash状态寄存器 (FLASH_STATUS)	110
11.11.3	Flash控制寄存器 (FLASH_CTRL)	110
11.11.4	Flash数据寄存器 (FLASH_DATA)	111
11.11.5	Flash地址寄存器 (FLASH_ADDR)	111
11.11.6	Flash Checksum寄存器 (FLASH_CHKSUM)	111
12	SERIAL-WIRE调试 (SWD)	112
12.1	概述	112
12.2	特性	112
12.3	引脚说明	112
12.4	调试注意事项	112
12.4.1	局限性	112
12.4.2	恢复调试功能	112
12.4.3	SWD引脚上的内部上拉/下拉电阻	112
13	开发工具	113
13.1	SN-LINK-V2/V3	113
13.2	SN32F268 STARTER-KIT	114
14	电气特性	115
14.1	极限参数	115
14.2	电气特性	115
15	FLASH ROM烧录引脚	116
16	封装信息	117
16.1	LQFP 48 PIN	117
16.2	QFN 46 PIN	118
16.3	SOP 28 PIN	119
16.4	SSOP 28 PIN	120
16.5	QFN 28 PIN	121
16.6	QFN 33 PIN	122
16.7	SSOP 24 PIN	123

17	芯片正印命名规则 .....	124
17.1	概述 .....	124
17.2	芯片型号说明 .....	124
17.3	命名举例 .....	125
17.4	日期码规则 .....	125

# 1 产品简介

## 1.1 功能特性

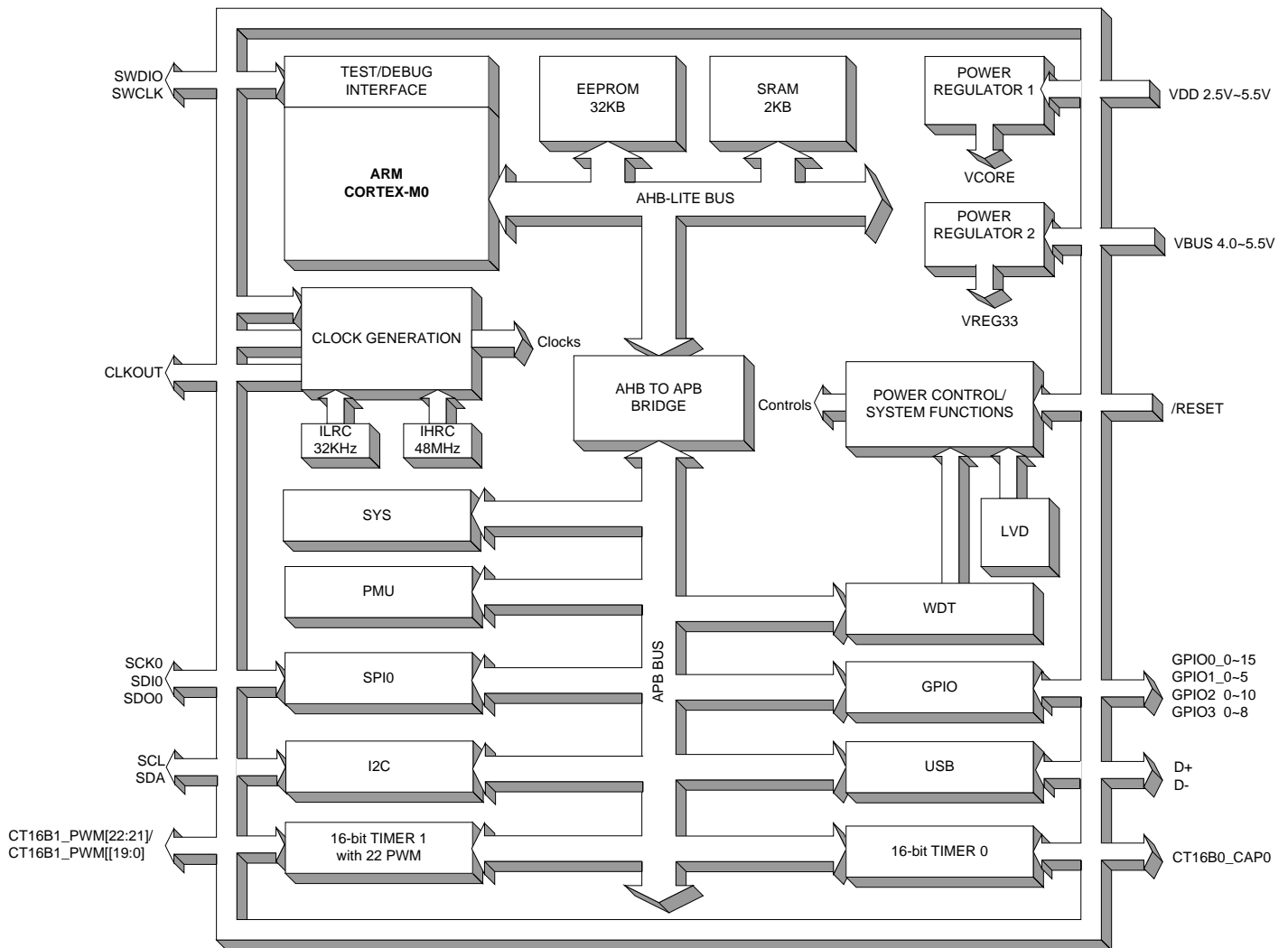
- ◆ **存储器配置**  
Flash ROM: 32KB (包括 2K Boot Loader)  
User RAM: 2KB  
USB FIFO RAM: 256 bytes
- ◆ **工作频率高达 48MHz**
- ◆ **中断源**  
ARM Cortex-M0 内建嵌套向量中断控制器 (NVIC)
- ◆ **IO 引脚配置**  
双向输入输出端口: P0、P1、P2、P3  
具有唤醒功能的端口: P0、P1、P2、P3 电平变换  
具有上拉电阻的端口: P0、P1、P2、P3  
20mA 灌电流/8mA 驱动电流端口: P0、P1、P2、P3
- ◆ **可编程的看门狗定时器 (WDT)**  
可编程看门狗频率和时钟源以及分频器
- ◆ **SysTick 定时器**  
24 位定时器  
系统时钟频率决定 SysTick 定时器时钟  
SysTick 定时器用于产生一个 10ms 的中断
- ◆ **LVD 支持不同的门限值**  
复位: 2.4V/3.3V for VDD
- ◆ **全速 USB 2.0**  
3.3V 校准输出, USB D+引脚内置 1.5K 上拉电阻。  
支持 1 个全速 USB 设备地址。  
支持 PS/2 模式  
1 个控制端点, 4 个可配置的 INT/Bulk 端点  
EP0 支持 64 字节 FIFO Depth。  
可编程的 EP1~EP4 FIFO Depth。  
共 5 个端点共用 256 字节 USB RAM。
- ◆ **工作电压: 2.5V ~ 5.5V**
- ◆ **定时器**  
1 个 16 位通用定时器 CT16B0 带 CAP0  
1 个 16 位通用定时器 CT16B1 带 22 路 PWM 输出
- ◆ **接口**  
1 个 I2C 控制器, 支持 I2C-bus 规格  
1 个 SPI 控制器, 支持 SPI 性能
- ◆ **2 系统时钟**  
内部高速时钟: RC, 48MHz。  
内部低速时钟: RC, 32KHz。
- ◆ **串行调试 (SWD)**
- ◆ **工作模式**  
普通模式, 睡眠模式和深度睡眠模式
- ◆ **FCPU (指令周期)**  
 $FCPU = FHCLK = FSYCLK / 1$ ,  $FSYCLK / 2$ ,  $FSYCLK / 4$ , ...,  $FSYCLK / 128$
- ◆ **支持在线烧录 (ISP)**
- ◆ **3.3V Regulator 输出**  
驱动 60mA 电流  
作为 USB D+内部上拉电阻的电源  
作为 P1.0~P1.5 (3.3V IO) 的 IO 电源  
作为外设 3.3V 的电压源
- ◆ **封装**  
LQFP48 pin  
QFN46 pin  
QFN33 pin  
QFN28 pin  
QFN28 pin  
SOP28/SSOP28 pin  
SSOP24 pin

☞ 产品性能表

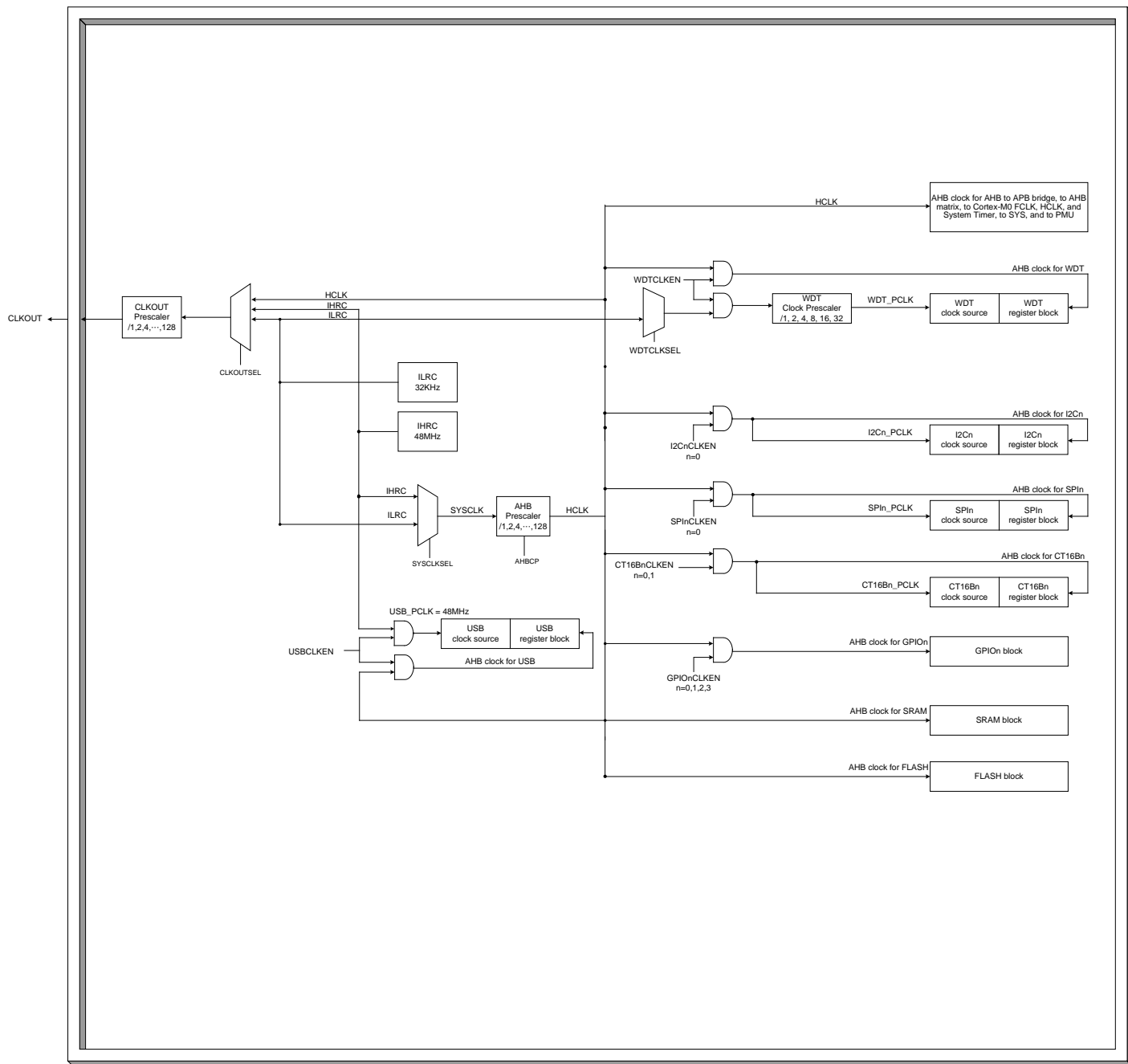
单片机名称	ROM	RAM	FCPU (Max MHz)	定时器	SPI	I2C	PWM	唤醒引 脚数目	封装
SN32F268F	32KB	2KB	48 MHz	16-bitx2	1	1	22-CH	42	LQFP48
SN32F267J	32KB	2KB	48 MHz	16-bitx2	1	1	22-CH	40	QFN46
SN32F265J	32KB	2KB	48 MHz	16-bitx2	1	1	17-CH	26	QFN33
SN32F2641J	32KB	2KB	48 MHz	16-bitx2	1	1	13-CH	22	QFN28
SN32F264S/X	32KB	2KB	48 MHz	16-bitx2	1	1	11-CH	22	SOP28/SSOP28
SN32F263X	32KB	2KB	48 MHz	16-bitx2	1	1	11-CH	18	SSOP24



## 1.2 系统框图

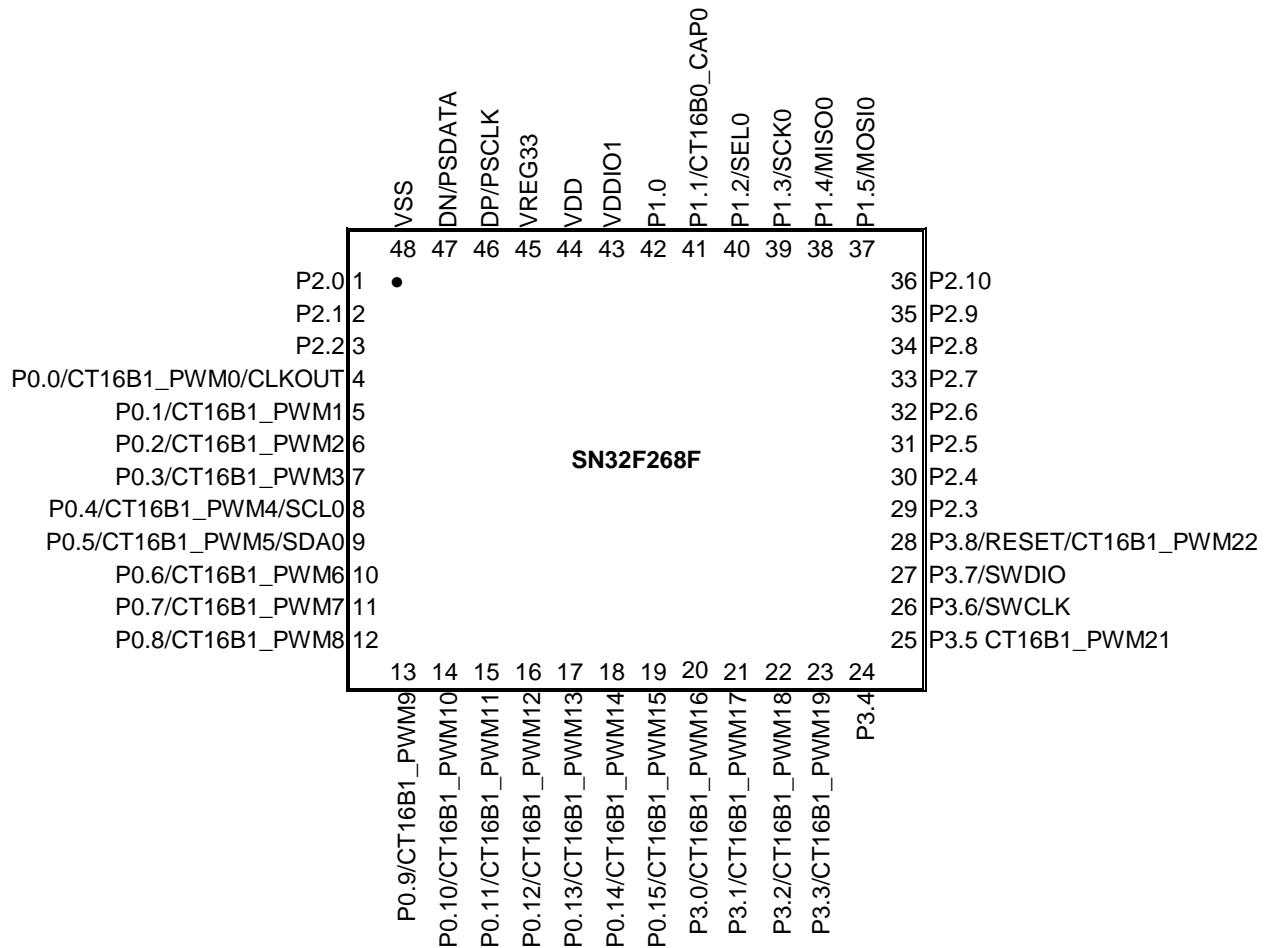


## 1.3 时钟产生框图

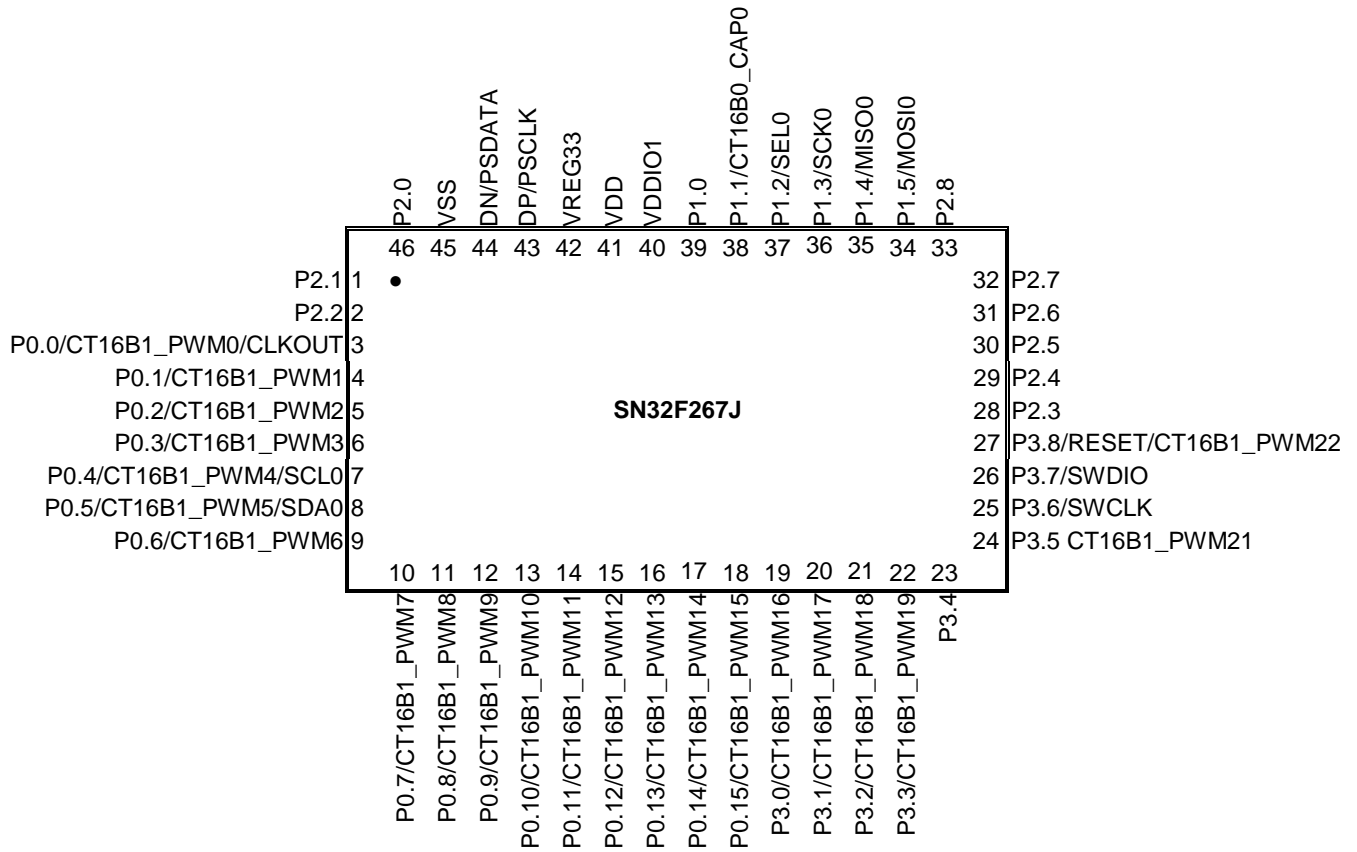


## 1.4 引脚配置

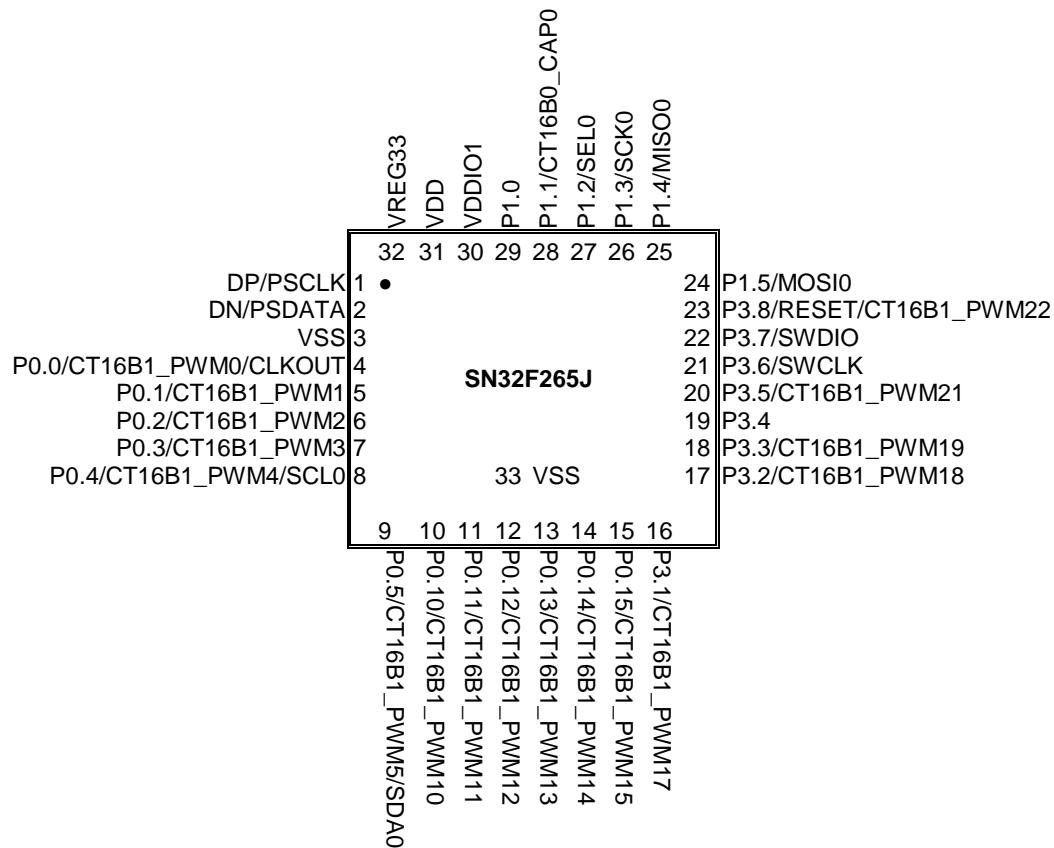
SN32F268F (LQFP 48 pins)



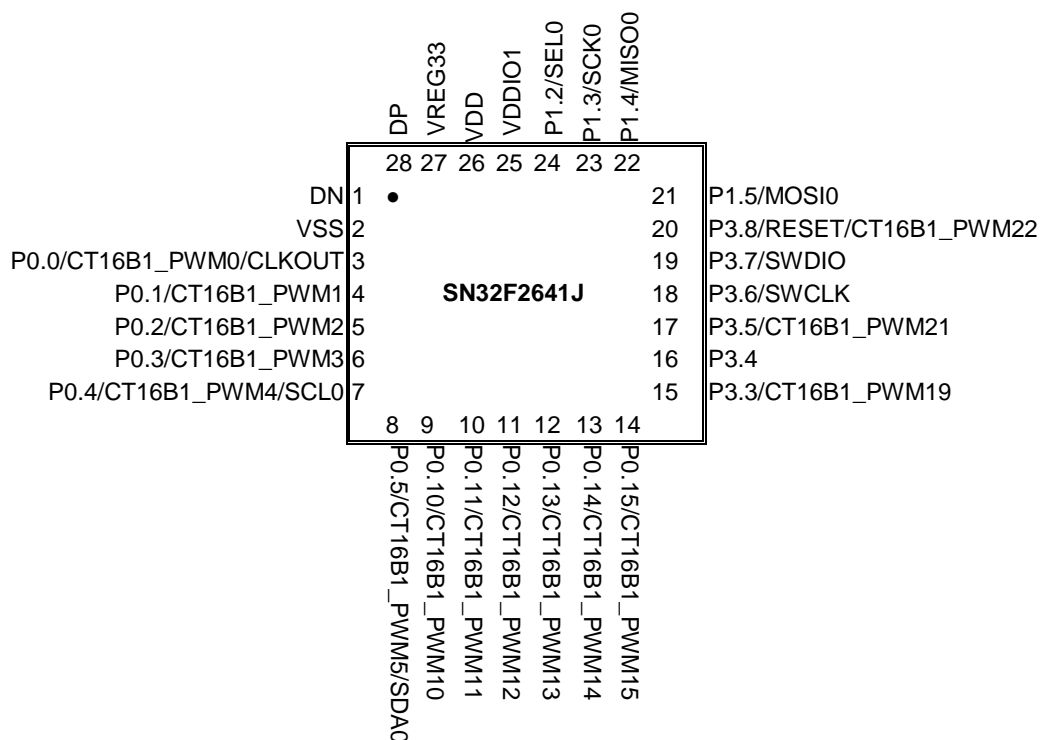
**SN32F267J (QFN 46 pins)**



**SN32F265J (QFN 33pins)**



**SN32F2641J (QFN 28pins)**



**SN32F264S/X (SOP/SSOP 28pins)**

VDDIO1	1	U	28	P1.0
VDD	2		27	P1.1/CT16B0_CAP0
VREG33	3		26	P1.2/SEL0
DP/PSCLK	4		25	P1.3/SCK0
DN/PSDATA	5		24	P1.4/MISO0
VSS	6		23	P1.5/MOSI0
P0.0/CT16B1_PWM0/CLKOUT	7		22	P3.8/RESET/CT16B1_PWM22
P0.1/CT16B1_PWM1	8		21	P3.7/SWDIO
P0.2/CT16B1_PWM2	9		20	P3.6/SWCLK
P0.3/CT16B1_PWM3	10		19	P3.5/CT16B1_PWM21
P0.4/CT16B1_PWM4/SCL0	11		18	P0.15/CT16B1_PWM15
P0.5/CT16B1_PWM5/SDA0	12		17	P0.14/CT16B1_PWM14
P0.10/CT16B1_PWM10	13		16	P0.13/CT16B1_PWM13
P0.11/CT16B1_PWM11	14		15	P0.12/CT16B1_PWM12

**SN32F264S/X**

**SN32F263X (SSOP 24pins)**

VDD	1	U	24	VDDIO1
VREG33	2		23	P1.1/CT16B0_CAP0
DP/PSCLK	3		22	P1.2/SEL0
DN/PSDATA	4		21	P1.3/SCK0
P0.0/CT16B1_PWM0/CLKOUT	5		20	P1.4/MISO0
P0.1/CT16B1_PWM1	6		19	P1.5/MOSI0
P0.2/CT16B1_PWM2	7		18	P3.8/RESET/CT16B1_PWM22
P0.3/CT16B1_PWM3	8		17	P3.7/SWDIO
P0.4/CT16B1_PWM4/SCL0	9		16	P3.6/SWCLK
P0.5/CT16B1_PWM5/SDA0	10		15	VSS
P0.10/CT16B1_PWM10	11		14	P0.13/CT16B1_PWM13
P0.11/CT16B1_PWM11	12		13	P0.12/CT16B1_PWM12

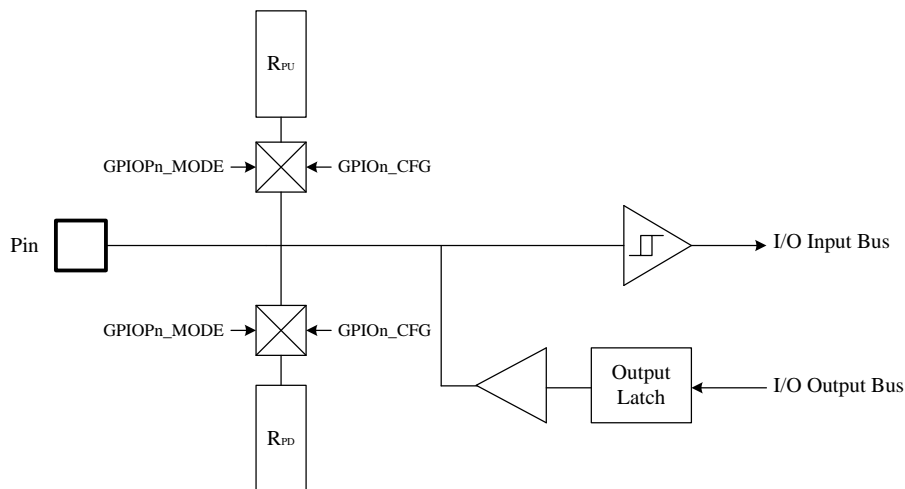
**SN32F263X**

## 1.5 引脚说明

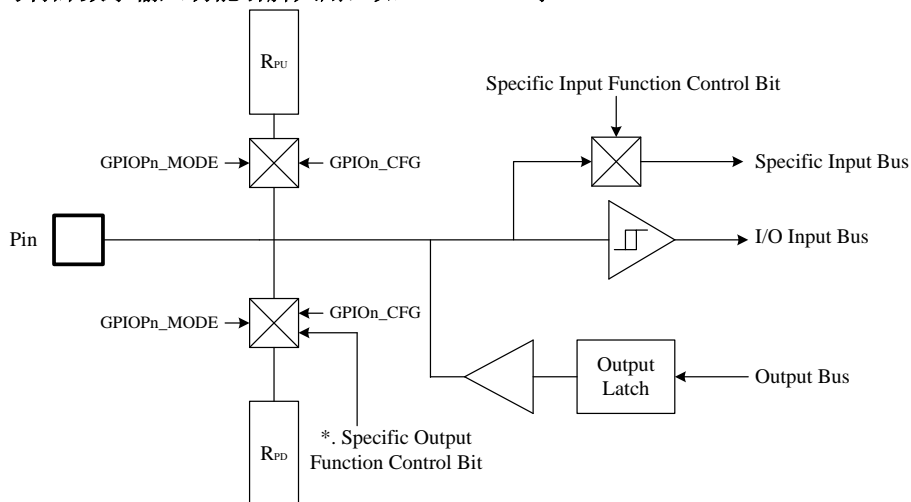
引脚名称	引脚类型	功能说明
VDD, VSS	P	数字电路电源输入端。
VREG33	O	3.3V 电源引脚。
DP/PSCLK	I/O	D+ —USB 差分信号线
	I/O	PSCLK — PS/2 时钟引脚，带 5K 上拉电阻。
DN/PSDATA	I/O	D- —USB 差分信号线
	I/O	PSDATA — PS/2 数据引脚，带 5K 上拉电阻。
VDDIO1	P	P1.0~P1.5 的 I/O 电源输入 Pad。
P0.0/CT16B1_PWM0/CLKOUT	I/O	P0.0 —双向输入输出引脚。
	O	CT16B1_PWM0 —CT16B1 PWM 输出引脚。
	O	CLKOUT — 时钟引脚。
P0.1/CT16B1_PWM1/PGDCLK	I/O	P0.1 —双向输入输出引脚。
	O	CT16B1_PWM1 —CT16B1 PWM 输出引脚。
	I/O	PGDCLK — 烧录模式下 Flash 时钟引脚。
P0.2/CT16B1_PWM2	I/O	P0.2 —双向输入输出引脚。
	O	CT16B1_PWM2 —CT16B1 PWM 输出引脚。
P0.3/CT16B1_PWM3	I/O	P0.3 —双向输入输出引脚。
	O	CT16B1_PWM3 —CT16B1 PWM 输出引脚。
P0.4/CT16B1_PWM4/SCL0	I/O	P0.4 —双向输入输出引脚。
	I/O	CT16B1_PWM4 —CT16B1 PWM 输出引脚。
	I/O	SCL0 — I2C clock input/output.
P0.5/CT16B1_PWM5/SDA0	I/O	P0.5 —双向输入输出引脚。
	I/O	CT16B1_PWM5 —CT16B1 PWM 输出引脚。
	I/O	SDA0 — I2C data input/output.
P0.6~P0.15/CT16B1_PWM6~15	I/O	P0.6~P0.15 —双向输入输出引脚。
	O	CT16B1_PWM6~15 —CT16B1 PWM 输出引脚。
P1.0	I/O	P1.0 —双向输入输出引脚。
P1.1/CT16B0_CAP0	I/O	P1.1 —双向输入输出引脚。
	I	CT16B0_CAP0 —CT16B0 捕捉输入引脚。
P1.2/SEL0	I/O	P1.2 —双向输入输出引脚。
	I	SEL0 —SPI 从动选择引脚。
P1.3/SCK0	I/O	P1.3 —双向输入输出引脚。
	I/O	SCK0 — SPI 串行时钟引脚。
P1.4/MISO0	I/O	P1.4 —双向输入输出引脚。
	I/O	MISO0 —SPI 主控输入从动输出引脚。
P1.5/MOSI0	I/O	P1.5 —双向输入输出引脚。
	I/O	MOSI0 —SPI 主控输出从动输入引脚。
P2.0~P2.10	I/O	P2.0~P2.10 —双向输入输出引脚。
P3.0~P3.3/CT16B1_PWM16~19	I/O	P3.0~P3.3 —双向输入输出引脚。
	O	CT16B1_PWM16~19 —CT16B1 PWM 输出引脚。
P3.4	I/O	P3.4 —双向输入输出引脚。
P3.5/CT16B1_PWM21	I/O	P3.5 —双向输入输出引脚。
	O	CT16B1_PWM21 —CT16B1 PWM 输出引脚。
P3.6/SWCLK	I/O	P3.6 —双向输入输出引脚。
	I	SWCLK —串行线时钟引脚。
P3.7/SWDIO	I/O	P3.7 —双向输入输出引脚。
	I/O	SWDIO —串行线数据输入/输出引脚。
P3.8/RESET/CT16B1_PWM22	I/O	P3.8 —双向输入输出引脚。
	I	RESET —外部复位引脚，施密特触发，低电平有效，通常保持高电平。
	O	CT16B1_PWM22 —CT16B1 PWM 输出引脚。

## 1.6 引脚电路结构图

- 双向 I/O 引脚:

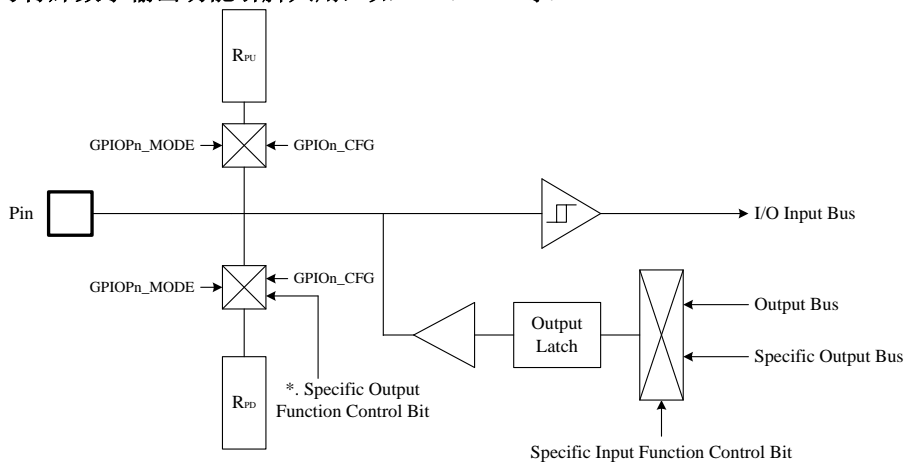


- 双向 I/O 引脚，与特殊数字输入功能引脚共用，如 SPI、I2C 等:



\*. Some specific functions switch I/O direction directly, not through GPIOFn\_MODE register.

- 双向 I/O 引脚，与特殊数字输出功能引脚共用，如 SPI、I2C 等:

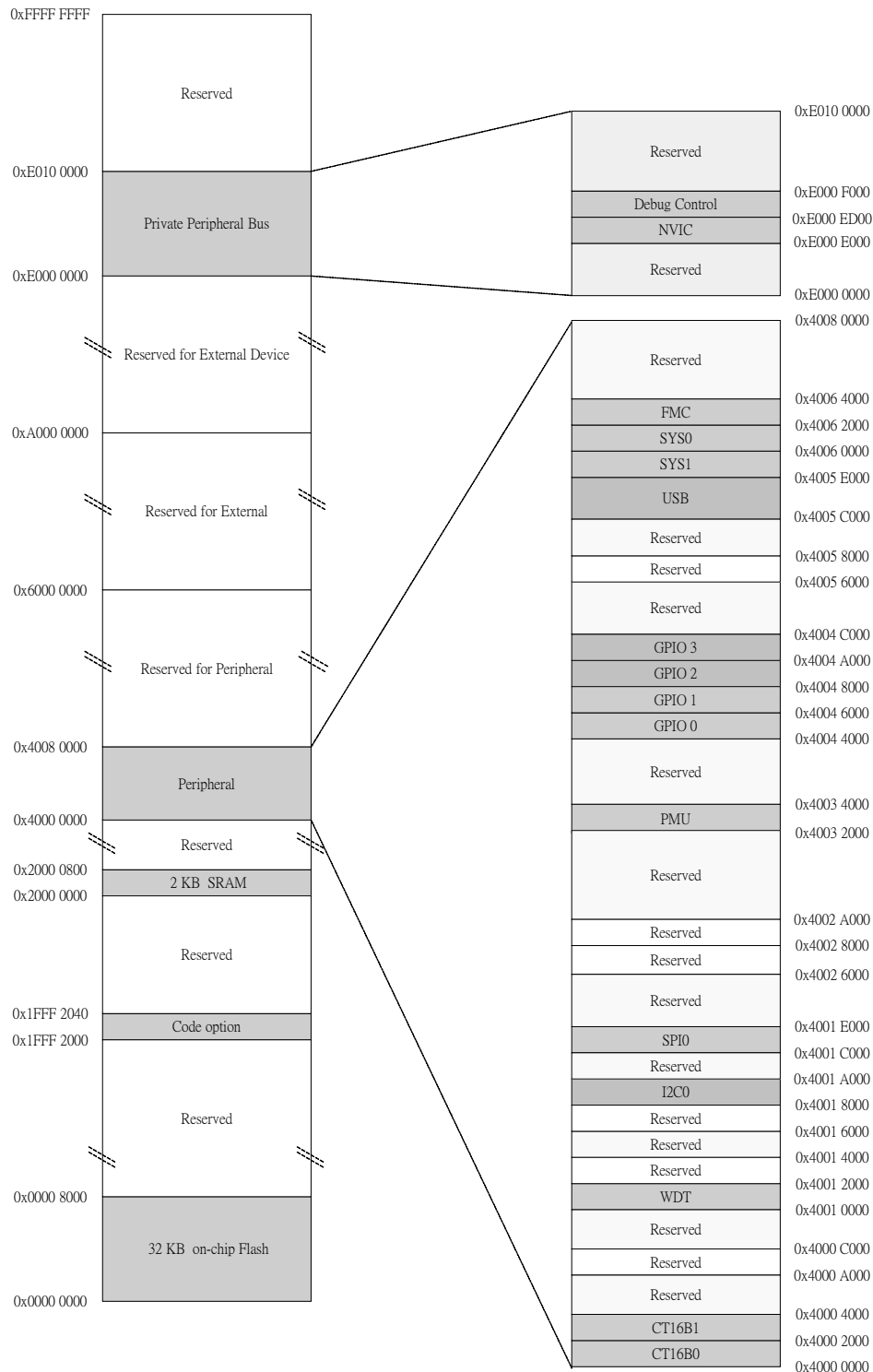


\*. Some specific functions switch I/O direction directly, not through GPIOFn\_MODE register.



# 2 中央处理器 (CPU)

## 2.1 存储器



## 2.2 系统TICK定时器

SysTick 定时器是 Cortex-M0 的一个集成部分，可以产生一个 10ms 的中断，用于操作系统或者其他系统软件管理。

由于 SysTick 定时器是 Cortex-M0 的一部分，通过提供一个标准定时器（基于 Cortex-M0 设备上有效）使软件端口更容易。

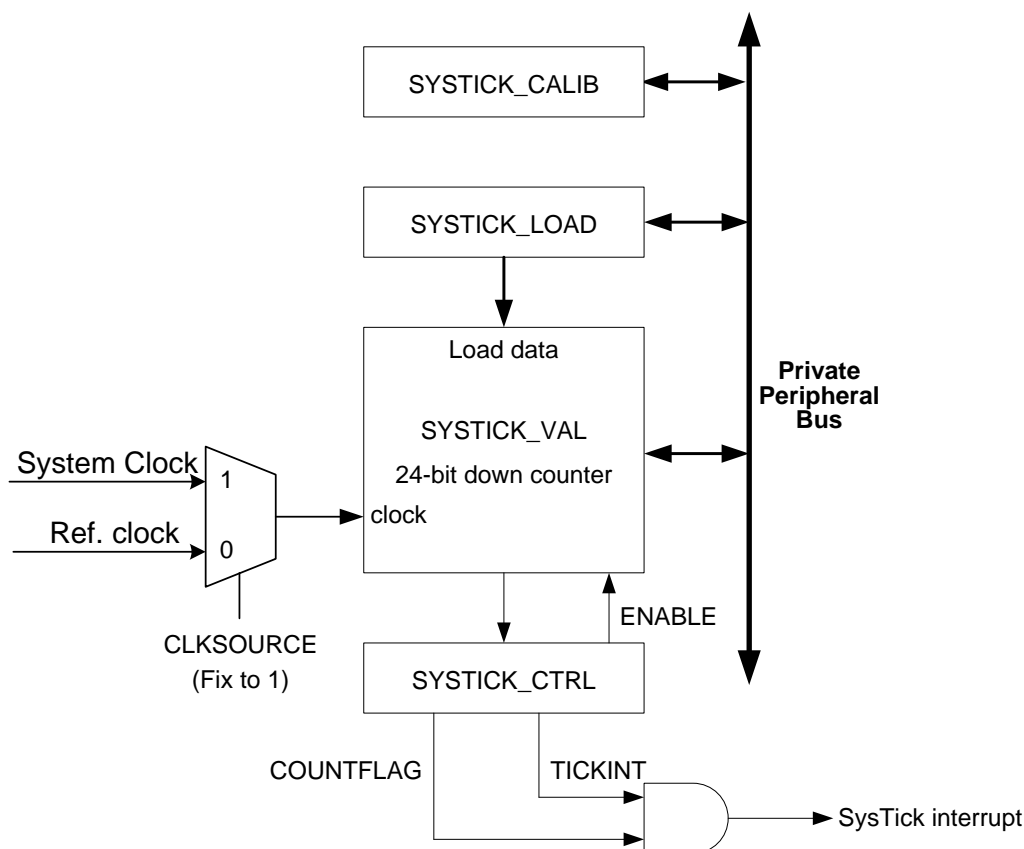
详情请参考 Cortex-M0 用户指南。

### 2.2.1 操作

SysTick 定时器是一个 24 位递减计数定时器，递减计数至 0 时产生中断。

SysTick 定时器由 SysTick 控制寄存器控制使能，时钟频率固定为系统时钟，它可以提供一个固定的 10ms 中断。

☞ SysTick 定时器的结构框图



当使能 SysTick 定时器，定时器从当前数值（SYSTICK\_VAL）开始递减计数至 0，在下一个时钟沿将重新载入 SYST\_RVR 的值到 SysTick 重装值寄存器（SYSTICK\_LOAD）中，然后在随后的时钟里递减计数。当计数器计数至 0 时，COUNTFLAG 状态位置 1，读取后该位清零。

\* 注：当处理器因调试而停止时，计数器不会进行递减计数。

### 2.2.2 SYSTICK用法说明及技巧

中断控制器时钟更新 SysTick 计数器。一些操作可以停止 SysTick 时钟信号以进入低功耗模式，此时 SysTick 计数器停止计数。

软件务必使用整字（word）方式操作 SysTick 寄存器。

SysTick 计数器重装数据和当前数据不会由 HW 来初始化。这意味着 SysTick 计数器正确的初始化流程如下：

1. 编程重装值到SYSTICK\_LOAD寄存器中。
2. 通过写入任意值到SYSTICK\_VAL寄存器，将当前值清除。
3. 设置控制和状态（SYSTICK\_CTRL）寄存器。

## 2.2.3 SYSTICK寄存器

### 2.2.3.1 系统Tick定时器控制和状态寄存器 (SYSTICK\_CTRL)

地址: 0xE000 E010 (参考 Cortex-M0 规格)

Bit	Name	Description	Attribute	Reset
31:17	Reserved		R	0
16	COUNTFLAG	SysTick 计数器递减计数到 0 时, 该位置 1, 被读取后清零。	R/W	0
15:3	Reserved		R	0
2	CLKSOURCE	选择 SysTick 定时器时钟源。 0: 参考时钟; 1: 系统时钟 (固定)。	R	1
1	TICKINT	SysTick 中断使能位。 0: 禁止 SysTick 中断。 1: 使能 SysTick 中断 (当 SysTick 计数器递减计数到 0 时产生中断)。	R/W	0
0	ENABLE	SysTick 计数器使能位。 0: 禁止; 1: 使能。	R/W	0

### 2.2.3.2 系统Tick定时器重装值寄存器 (SYSTICK\_LOAD)

地址: 0xE000 E014 (参考 Cortex-M0 规格)

重装寄存器需设置当 SysTick 递减计数到 0 时, 重新装载的数据。此寄存器由软件来初始化。如果系统或外部时钟运行在预期的频率, SYSTICK\_LOAD 寄存器的值可以使用作为重装寄存器的值。

下面的示例说明选择 SysTick 定时器重装值, 以获得 10ms 的间隔时间, 系统时钟设置为 48MHz。

**SysTick 时钟 = 系统时钟 = 48 MHz**

**RELOAD = (SysTick 时钟频率 × 10 ms) - 1 = (48 MHz × 10 ms) - 1 = 0x000752FF**

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	RELOAD	使能计数器且计数到 0 时, 此值重新装入 SYSTICK_LOAD 寄存器。	R/W	0x5F7F9B

### 2.2.3.3 系统Tick定时器当前值寄存器 (SYSTICK\_VAL)

地址: 0xE000 E018 (Refer to Cortex-M0 Spec)

Bit	Name	Description	Attribute	Reset
31:24	Reserved		R	0
23:0	CURRENT	读取该寄存器时返回 SysTick 计数器的当前值; 写入任何数据清除 SysTick 计数器和 SYSTICK_CTRL 寄存器中的 COUNTFLAG 位。	R/W	0x7E7F35

### 2.2.3.4 系统Tick定时器校准值寄存器 (SYSTICK\_CALIB)

地址: 0xE000 E01C (参考 Cortex-M0 规格)

Bit	Name	Description	Attribute	Reset
31	NOREF	显示是否提供参考时钟给 M0。 1: 没有提供参考时钟	R	1
30	SKEW	显示 TENMS 的值是否准确, 一个不准确的 TENMS 值会影响 SysTick 作为软件实时时钟的匹配度。 0: TENMS 的值是准确的; 1: TENMS 的值不准确, 或者不存在。	R	0
29:24	Reserved		R	0
23:0	TENMS	为 10ms 定时作用的重装数值, 同时受系统时钟偏差影响。如果这个值读出为 0, 则此校准值是不确定的	R/W	0xA71FF

## 2.3 嵌套向量中断控制器（NVIC）

包括内核异常在内的所有中断，都由 NVIC 来管理。NVIC 的特性如下：

1. NVIC 支持 32 个向量中断；
2. 带有硬件掩膜优先级的 4 级可编程中断优先级；
3. 快速响应异常情形和中断处理；
4. 对延迟中断的有效处理；
5. 对系统控制寄存器的操作；
6. 产生软件中断。

### 2.3.1 中断和异常向量

Execution No.	Priority	Function	Description	Address Offset
0	-	-	Reserved	0x0000 0000
1	-3	Reset	Reset	0x0000 0004
2	-2	NMI_Handler	Non maskable interrupt.	0x0000 0008
3	-1	HardFault_Handler	All class of fault	0x0000 000C
4~10	Reserved	Reserved	Reserved	-
11	Settable	SVCCall		0x0000 002C
12~13	Reserved	Reserved	Reserved	-
14	Settable	PendSV		0x0000 0038
15	Settable	SysTick		0x0000 003C
16	Settable	IRQ0/NDTIRQ	NDT	0x0000 0040
17	Settable	IRQ1/USBIRQ	USB	0x0000 0044
18	Settable	IRQ2/		0x0000 0048
19	Settable	IRQ3/		0x0000 004C
20	Settable	IRQ4/		0x0000 0050
21	Settable	IRQ5/		0x0000 0054
22	Settable	IRQ6/		0x0000 0058
23	Settable	IRQ7/		0x0000 005C
24	Settable	IRQ8/		0x0000 0060
25	Settable	IRQ9/		0x0000 0064
26	Settable	IRQ10/		0x0000 0068
27	Settable	IRQ11/		0x0000 006C
28	Settable	IRQ12/		0x0000 0070
29	Settable	IRQ13/SPI0IRQ	SPI0	0x0000 0074
30	Settable	IRQ14/		0x0000 0078
31	Settable	IRQ15/ I2C0IRQ	I2C0	0x0000 007C
32	Settable	IRQ16/CT16B0IRQ	CT16B0	0x0000 0080
33	Settable	IRQ17/CT16B1IRQ	CT16B1	0x0000 0084
34	Settable	IRQ18/		0x0000 0088
35	Settable	IRQ19/		0x0000 008C
36	Settable	IRQ20/		0x0000 0090
37	Settable	IRQ21/		0x0000 0094
38	Settable	IRQ22/		0x0000 0098
39	Settable	IRQ23/		0x0000 009C
40	Settable	IRQ24/		0x0000 00A0
41	Settable	IRQ25/WDTIRQ	WDT	0x0000 00A4
42	Settable	IRQ26/LVDIRQ	LVD	0x0000 00A8
43	Settable	IRQ27/		0x0000 00AC
44	Settable	IRQ28/P3IRQ	GPIO interrupt status of port 3	0x0000 00B0
45	Settable	IRQ29/P2IRQ	GPIO interrupt status of port 2	0x0000 00B4
46	Settable	IRQ30/P1IRQ	GPIO interrupt status of port 1	0x0000 00B8
47	Settable	IRQ31/P0IRQ	GPIO interrupt status of port 0	0x0000 00BC

## 2.3.2 NVIC寄存器

### 2.3.2.1 IRQ0~31 中断设置使能寄存器 (NVIC\_ISER)

地址: 0xE000 E100 (参考 Cortex-M0 规格)

ISER 使能中断, 并显示使能的中断。

Bit	Name	Description	Attribute	Reset
31:0	SETENA[31:0]	中断设置使能位。 写→0: 无影响; 1: 使能中断。 读→0: 禁止中断; 1: 使能中断。	R/W	0

### 2.3.2.2 IRQ0~31 中断清零使能寄存器 (NVIC\_ICER)

地址: 0xE000 E180 (参考 Cortex-M0 规格)

ICER 禁止中断, 并显示使能的中断。

Bit	Name	Description	Attribute	Reset
31:0	CLRENA[31:0]	中断清零使能位。 写→0: 无影响; 1: 禁止中断。 读→0: 禁止中断; 1: 使能中断。	R/W	0

### 2.3.2.3 IRQ0~31 中断挂起设置寄存器 (NVIC\_ISPR)

地址: 0xE000 E200 (参考 Cortex-M0 规格)

ISPR 迫使中断进入挂起状态, 并显示正在挂起的中断。

\* 注: 写 1 到 ISPR 位需符合:  
1、对正处于挂起状态的中断请求无影响;  
2、禁止的中断设置其状态为挂起状态。

Bit	Name	Description	Attribute	Reset
31:0	SETPEND[31:0]	中断挂起设置位。 写→0: 无影响; 1: 改变中断状态为挂起状态 读→0: 中断未处于挂起状态; 1: 中断处于挂起状态	R/W	0

### 2.3.2.4 IRQ0~31 中断挂起清零寄存器 (NVIC\_ICPR)

地址: 0xE000 E280 (参考 Cortex-M0 规格)

ICPR 可以取消中断的挂起状态, 并显示正在挂起的中断。

\* 注: 写入 1 到 ICPR, 并不影响相应中断的有效状态。

Bit	Name	Description	Attribute	Reset
31:0	CLRPEND[31:0]	中断挂起清除位。 写→0: 无影响; 1: 禁止中断的挂起状态。 读→0: 中断没有处于挂起状态; 1: 中断处于挂起状态。	R/W	0

**2.3.2.5 IRQ0~31 中断优先级寄存器 (NVIC\_IPRn) (n=0~7)**

地址: 0xE000 E400 + 0x4 \* n (参考 Cortex-M0 规格)

中断优先级寄存器为每个中断提供 8-bit 优先权选择级别, 每个寄存器管理 4 个中断源的优先权。这意味着寄存器的数量是明确定义的, 且与执行的中断的数量相对应。

Bit	Name	Description	Attribute	Reset
31:24	PRI_(4*n+3)	每个优先级字段的优先级设定值为 0-192。值越小, 对应中断的优先级越高。处理器只处理每个优先级字段的 bits[31:30], 而 bits[29:24]读出于 0 且写入无效, 因此如果写 255 到优先级寄存器, 寄存器只保存为 192	R/W	0
23:16	PRI_(4*n+2)	每个优先级字段的优先级设定值为 0-192。值越小, 对应中断的优先级越高。处理器只处理每个优先级字段的 bits[23:22], 而 bits[21:16]读出于 0 且写入无效, 因此如果写 255 到优先级寄存器, 寄存器只保存为 192	R/W	0
15:8	PRI_(4*n+1)	每个优先级字段的优先级设定值为 0-192。值越小, 对应中断的优先级越高。处理器只处理每个优先级字段的 bits[15:14], 而 bits[13:8]读出于 0 且写入无效, 因此如果写 255 到优先级寄存器, 寄存器只保存为 192	R/W	0
7:0	PRI_4*n	每个优先级字段的优先级设定值为 0-192。值越小, 对应中断的优先级越高。处理器只处理每个优先级字段的 bits[7:6], 而 bits[5:0]读出于 0 且写入无效, 因此如果写 255 到优先级寄存器, 寄存器只保存为 192	R/W	0

## 2.4 应用中断和复位控制（AIRC）

地址：0xE000 ED0C（参考 Cortex-M0 规格）

整个 MCU，包括内核，都可以通过软件设置 AIRC 寄存器的 SYSRESREQ 位来复位。

\* 注：在写入数据到该寄存器时，用户必须同时写入 0x05FA 到 VECTKEY，否则处理器会忽略此次写动作。

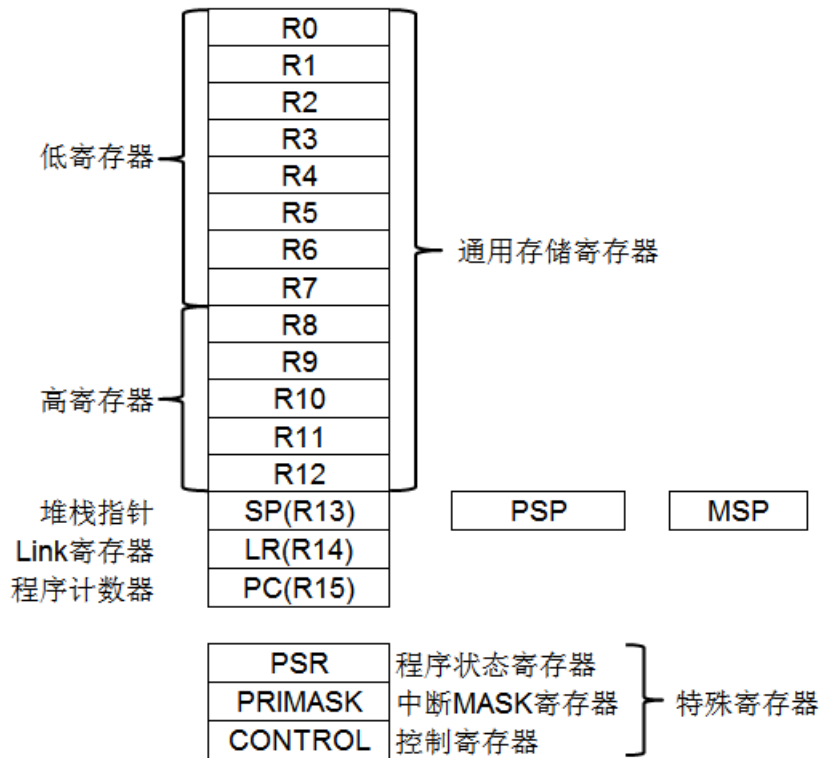
Bit	Name	Description	Attribute	Reset
31:16	VECTKEY	寄存器关键字。 读取为未知数；写入 0x05FA 到 VECTKEY，否则写动作无效。	R/W	0
15	ENDIANESS	数据字节顺序应用位。 0：小端排列；1：大端排列。	R	0
14:3	Reserved		R	0
2	SYSRESETREQ	系统复位请求位，该位读取为 0。 0：没有影响；1：请求系统复位。	W	0
1	VECTCLRACTIVE	保留以用来调试，该位读取为 0。写入数据到寄存器时，用户必须写入 0 到该位，否则会发生未知情形。	W	0
0	Reserved		R	0

## 2.5 编译选项列表（CODE OPTION）

地址：0x1FFF 2000

Bit	Name	Description	Attribute	Reset
31:16	Code Security[15:0]	代码加密。 0xFFFF：CS0；0x5A5A：CS1；0xA5A5：CS2；Other：CS2。	R/W	0xFFFF
15:0	Reserved		R	All 1

## 2.6 核心寄存器简介



Register	Description（参考 Cortex-M0 规格）																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
R0~R12	通用存储寄存器，用于存储操作数据。																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
SP (R13)	堆栈指针（SP）。在 thread 模式下，CONTROL 寄存器指示堆栈指针的使用，主堆栈指针（MSP）或者处理堆栈指针（PSP）。 在复位以后，处理器装载地址 0x00000000 处的值到 MSP。																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
LR (R14)	Link 寄存器（LR）。存储子程序，CALL 功能及其它异常的返回信息。																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
PC (R15)	程序计数器（PC）。包含了当前程序地址。 复位时，处理器装载地址 0x00000004 的复位向量值到 PC。																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
PSR	<p>程序状态寄存器（PSR）包含了下面几种： 应用程序状态寄存器（APSR）； 中断程序状态寄存器（IPSR）； 执行程序状态寄存器（EPSR）。 这些寄存器在 32 位 PSR 中是相互独立的。</p> <table><tr><td></td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>25</td><td>24</td><td>23</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>&lt;/</td></tr></table>		31	30	29	28	27	25	24	23																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												</
	31	30	29	28	27	25	24	23																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												</		



## 3

## 系统控制

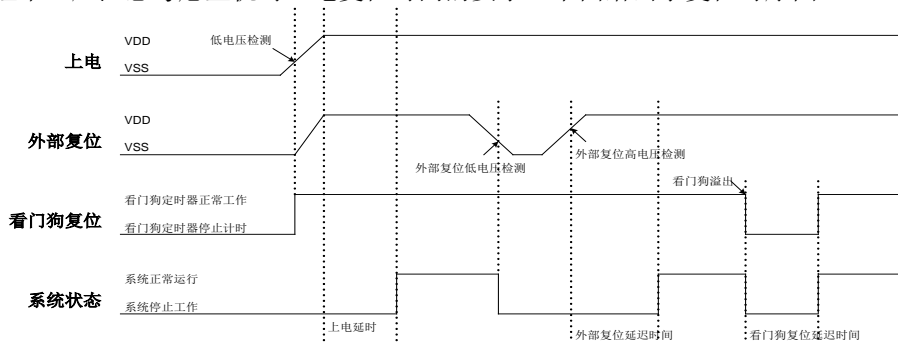
## 3.1 复位

SN32F260 系列单片机的复位方式有以下几种：

- 低电平到 RST 引脚触发复位（外部复位）；
- 上电复位（POR 复位）
- LVD 复位；
- 看门狗定时器复位（WDT 复位）；
- 软件复位（SW 复位）；

系统复位源可通过系统复位状态寄存器（SYS0\_RSTST）的不同标志位来识别。复位源作用于 RST 引脚，且 RST 引脚在延迟阶段保持低电平。复位程序的复位向量地址位于储存器的 0x00000004 处。详情请参考中断和 Exception 向量。

系统复位需要一定的时间，并提供完整的上电复位规程。对于不同类型的振荡器，完成复位所需要的时间也不同。因此，VDD 的上升速度和不同晶振的起振时间都不固定。RC 振荡器的启动时间非常短，晶体振荡器的启动时间则比较长。在用户终端使用的过程中，应注意考虑主机对上电复位时间的要求。下面给出了复位时序图。



## 3.1.1 上电复位（POR）

上电复位与 LVD 操作密切相关。系统上电过程呈逐渐上升的曲线形式，需要一定时间才能达到正常电平值。上电复位的时序如下：

- 上电：系统检测到电源电压上升并等待其稳定；
- 外部复位（使能外部复位引脚时才有效）：系统检测外部复位引脚状态。如果不为高电平，系统保持复位状态直到外部复位引脚的复位结束。
- 系统初始化：所有的系统寄存器被置为默认状态；
- 振荡器开始工作：振荡器开始提供系统时钟；
- 执行程序：上电结束后，程序从引导装载程序处开始运行。

## 3.1.2 看门狗复位（WDT RESET）

看门狗复位是系统的一种保护设置。在正常状态下，由程序将看门狗定时器清零。若出错，系统处于未知状态，程序无法清除看门狗，导致看门狗定时器溢出，此时系统复位。看门狗复位后，系统重启进入正常状态。看门狗复位的时序如下：

- 看门狗定时器状态：系统检测看门狗定时器是否溢出，若溢出，则系统复位；
- 系统初始化：初始化所有的系统寄存器；
- 振荡器起振：振荡器正常工作并提供系统时钟；
- 执行程序：上电复位结束，程序从地址 0x00 开始执行。

看门狗定时器应用时需注意：

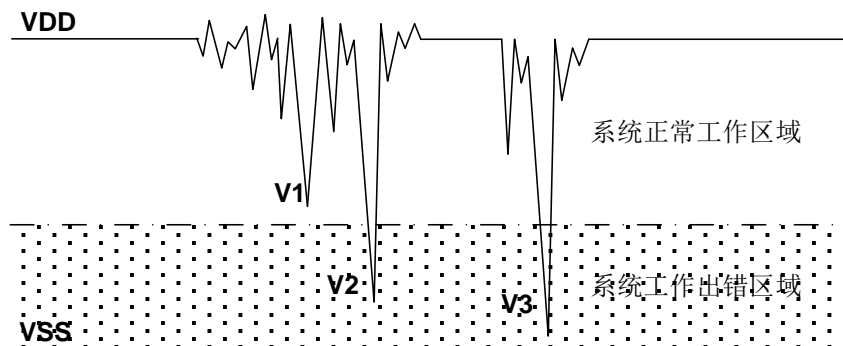
- 清看门狗定时器之前，请先检查 I/O 口的状态和 RAM 数据，可降低系统错误率；
- 不能在中断里清看门狗定时器，否则无法起到侦测程序跑飞的目的；
- 程序中应该只有一个清看门狗的动作，这种架构能够最大限度的发挥看门狗的保护功能。

\* 注：关于看门狗定时器的详细内容，请参阅看门狗定时器有关章节。

### 3.1.3 掉电复位

#### 3.1.3.1 掉电复位简述

掉电复位针对外部因素引起的系统电压跌落情形（例如：干扰或外部负载的变化），掉电复位可能会引起系统工作状态不正常或程序执行错误。



掉电复位示意图

电压跌落可能会进入系统死区。系统死区意味着电源不能满足系统的最小工作电压要求。上图是一个典型的掉电复位示意图。图中，VDD 受到严重的干扰，电压值降的非常低。虚线以上区域系统正常工作，在虚线以下的区域内，系统进入未知的工作状态，这个区域称作死区。当 VDD 跌至 V1 时，系统仍处于正常状态；当 VDD 跌至 V2 和 V3 时，系统进入死区，则容易导致出错。以下情况系统可能进入死区：

##### DC 运用中：

DC 运用中一般都采用电池供电，当电池电压过低或单片机驱动负载时，系统电压可能跌落并进入死区。这时，电源不会进一步下降到 LVD 检测电压，因此系统维持在死区。

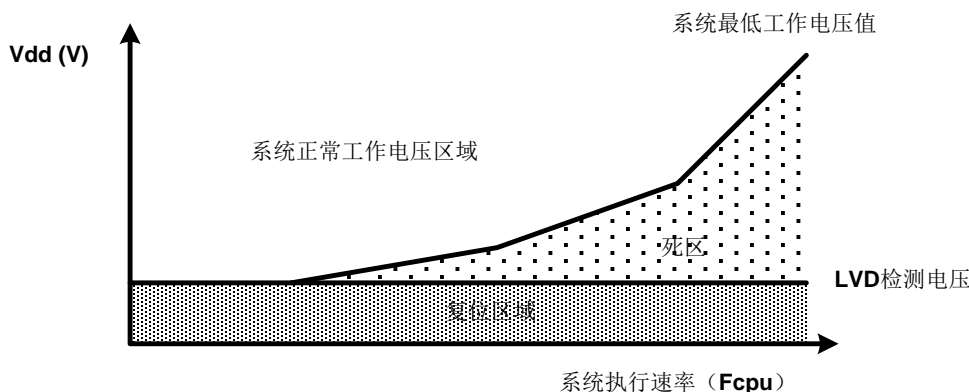
##### AC 运用中：

系统采用 AC 供电时，DC 电压值受 AC 电源中的噪声影响。当外部负载过高，如驱动马达时，负载动作产生的干扰也影响到 DC 电源。VDD 若由于受到干扰而跌落至最低工作电压以下时，则系统将有可能进入不稳定工作状态。

在 AC 运用中，系统上、下电时间都较长。其中，上电时序保护使得系统正常上电，但下电过程却和 DC 运用中情形类似，AC 电源关断后，VDD 电压在缓慢下降的过程中易进入死区。

#### 3.1.3.2 系统工作电压

为了改善系统掉电复位的性能，首先必须明确系统具有的最低工作电压值。系统最低工作电压与系统执行速度有关，不同的执行速度下最低工作电压值也不同。



系统工作电压与执行速度关系图

如上图所示，系统正常工作电压区域一般高于系统复位电压，同时复位电压由低电压检测（LVD）电平决定。当系统执行速度提高时，系统最低工作电压也相应提高，但由于系统复位电压是固定的，因此在系统最低工作电压与系统复位电压之间就会出现一个电压区域，系统不能正常工作，也不会复位，这个区域即为死区。

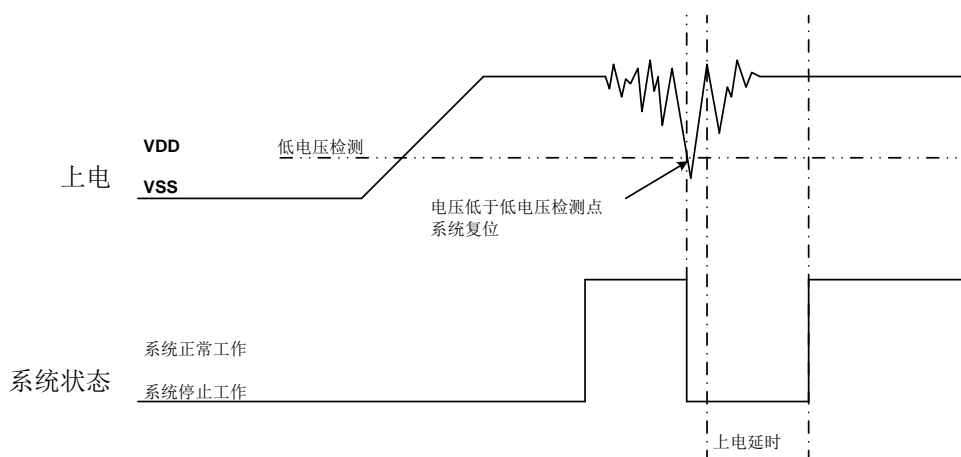
### 3.1.3.3 掉电复位性能改进

如何改善系统掉电复位性能，有以下几点建议：

- LVD 复位；
- 看门狗复位；
- 降低系统工作速度；
- 采用外部复位电路（稳压二极管复位电路，电压偏移复位电路，外部 IC 复位）。

\* 注：“稳压二极管复位电路”、“电压偏移复位电路”和“外部 IC 复位”能够完全避免掉电复位出错。

**LVD 复位：**



低电压检测（LVD）是 SONiX 32 位单片机内置的掉电复位保护装置，当 VDD 跌落并低于 LVD 检测电压值时，LVD 给出一个中断信号到 NVIC，以便触发 CPU 中断；若不能触发中断，则 SW 通过读取寄存器的显示状态来监控该信号。可以选择一个附加的极限值强制系统复位。不同的单片机有不同的 LVD 检测电平，LVD 检测电平值仅为一个电压点，并不能覆盖所有死区范围。因此采用 LVD 依赖于系统要求和环境状况。电源变化较大时，LVD 能够起到保护作用，如果电源变化触发 LVD，系统工作仍出错，那么 LVD 就不能起到保护作用，就需要采用其它复位方法。更多 LVD 信息请参考电气特性。

**看门狗复位：**

看门狗定时器用于保证系统正常工作。通常，会在主程序中将看门狗定时器清零，但不要在多个分支程序中清看门狗。若程序正常运行，看门狗不会复位。当系统进入死区或程序运行出错的时候，看门狗定时器继续计数直至溢出，系统复位。如果看门狗复位后电源仍处于死区，则系统复位失败，保持复位状态，直到系统工作电压恢复到正常值。

**降低系统工作速度：**

系统工作速度越快最低工作电压值越高，从而加大工作死区的范围，因此降低系统工作速度不失为降低系统进入死区几率的有效措施。所以，可选择合适的工作速度以避免系统进入死区，这个方法需要调整整个程序使其满足系统要求。

**附加外部复位电路：**

外部复位也能够完全改善掉电复位性能。有三种外部复位方式可改善掉电复位性能：稳压二极管复位电路，电压偏移复位电路和外部 IC 复位。它们都采用外部复位信号控制单片机可靠复位。

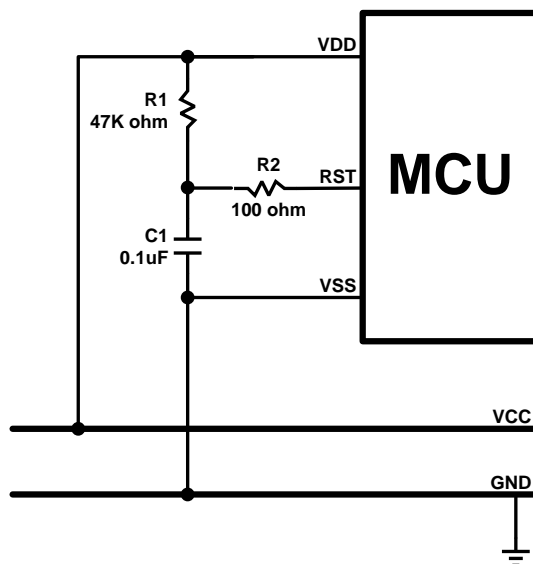
### 3.1.4 外部复位

外部复位功能由外部复位引脚控制寄存器（SYS0\_EXRSTCTRL）控制，初始值为 1，即使能外部复位功能。外部复位引脚为施密特触发结构，低电平有效。复位引脚处于高电平时，系统正常运行。当复位引脚输入低电平信号时，系统复位。外部复位操作在上电和正常工作模式时有效。需要注意的是，在系统上电完成后，外部复位引脚必须输入高电平，否则系统将一直保持在复位状态。外部复位的时序如下：

- 外部复位（当且仅当外部复位引脚为使能状态）：系统检测复位引脚的状态，如果复位引脚不为高电平，则系统会一直保持在复位状态，直到外部复位结束；
- 系统初始化：所有的系统寄存器被置为初始状态；
- 振荡器开始工作：振荡器开始提供系统时钟；
- 执行程序：上电结束后，程序从引导装载程序处开始运行。

外部复位可以在上电过程中使系统复位。良好的外部复位电路可以保护系统以免进入未知的工作状态，如 AC 应用中的掉电复位等。

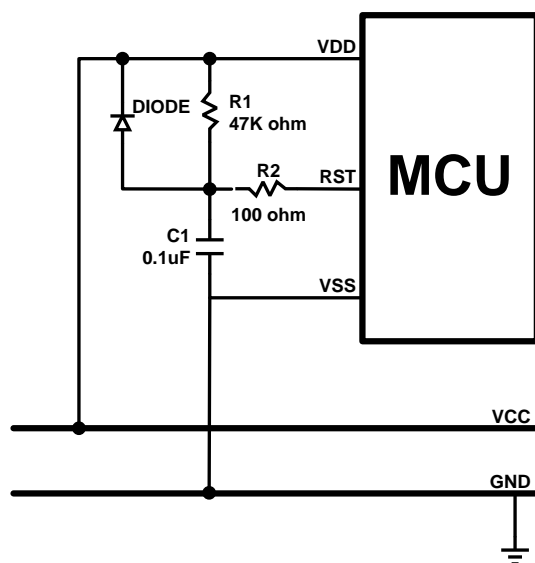
#### 3.1.4.1 RC复位电路



上图为一个由电阻 R1 和电容 C1 组成的基本 RC 复位电路，它在系统上电的过程中能够为复位引脚提供一个缓慢上升的复位信号。这个复位信号的上升速度低于 VDD 的上电速度，为系统提供合理的复位时序，当复位引脚检测到高电平时，系统复位结束，进入正常工作状态。

\* 注：此 RC 复位电路不能解决非正常上电和掉电复位问题。

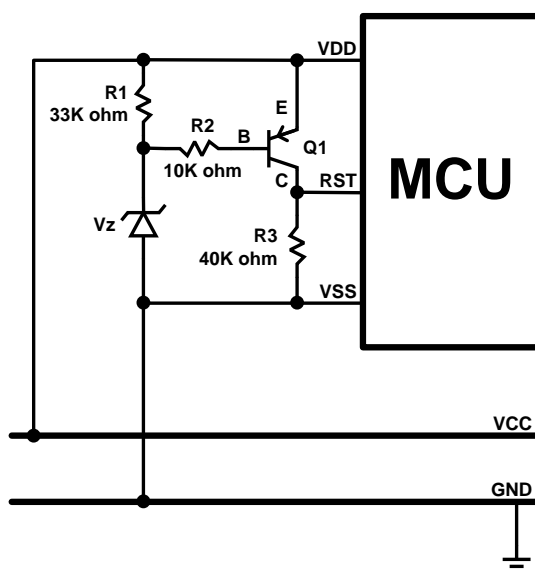
## 3.1.4.2 二极管&amp;RC复位电路



上图中，R1 和 C1 同样是为复位引脚提供输入信号。对于电源异常情况，二极管正向导通使 C1 快速放电并与 VDD 保持一致，避免复位引脚持续高电平、系统无法正常复位。

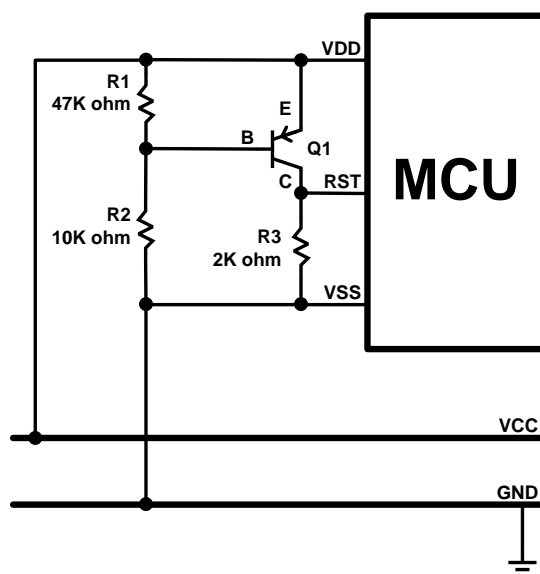
\* 注：“基本 RC 复位电路”和“二极管及 RC 复位电路”中的电阻 R2 都是必不可少的限流电阻，以避免复位引脚 ESD (Electrostatic Discharge) 或 EOS (Electrical Over-stress) 击穿。

## 3.1.4.3 稳压二极管复位电路



稳压二极管复位电路是一种简单的 LVD 电路，基本上可以完全解决掉电复位问题。如上图电路中，利用稳压管的击穿电压作为电路复位检测值，当 VDD 高于 “ $V_z + 0.7V$ ” 时，三极管集电极输出高电平，单片机正常工作；当 VDD 低于 “ $V_z + 0.7V$ ” 时，三极管集电极输出低电平，单片机复位。稳压管规格不同则电路复位检测值不同，根据电路的要求选择合适的二极管。

## 3.1.4.4 电压偏置复位电路

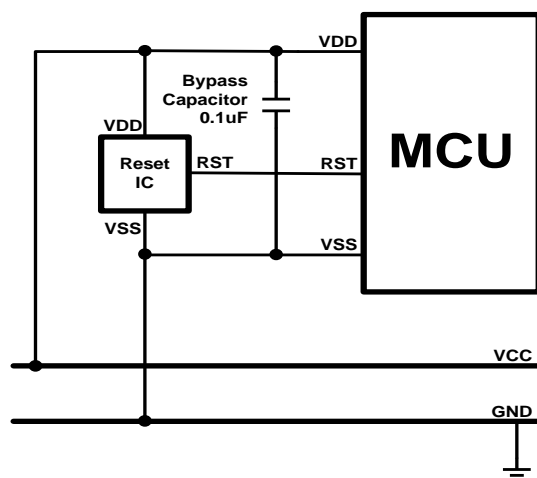


电压偏置复位电路是一种简单的 LVD 电路，基本上可以完全解决掉电复位问题。与稳压二极管复位电路相比，这种复位电路的检测电压值的精确度有所降低。电路中，R1 和 R2 构成分压电路，当 VDD 高于和等于分压值  $0.7V \times (R1 + R2) / R1$  时，三极管集电极 C 输出高电平，单片机正常工作；VDD 低于  $0.7V \times (R1 + R2) / R1$  时，集电极 C 输出低电平，单片机复位。

对于不同应用需求，选择适当的分压电阻。单片机复位引脚上电压的变化与 VDD 电压变化之间的差值为 0.7V。如果 VDD 跌落并低于复位引脚复位检测值，那么系统将被复位。如果希望提升电路复位电平，可将分压电阻设置为  $R2 > R1$ ，并选择 VDD 与集电极之间的结电压高于 0.7V。分压电阻 R1 和 R2 在电路中要耗电，此处的功耗必须计入整个系统的功耗中。

\* 注：在电源不稳定或掉电复位的情况下。“稳压二极管复位电路”和“偏压复位电路”能够保护电路在电压跌落时避免系统出错。当电压跌落至低于复位检测值时，系统将被复位。从而保证系统正常工作。

## 3.1.4.5 外部IC复位电路



外部 IC 复位可以增强 MCU 复位的可靠性。

### 3.1.5 软件复位

整个单片机，包括内核部分，都可以由软件通过设置 SHYSRESREQ 位（在 Cortex-M0 规格中位于 AIRC—应用中断和复位控制寄存器中）来复位。

软件初始化系统复位的流程如下：

- 1、通过设置 SYSRESREQ 位初始化；
- 2、宣告内部复位；
- 3、内部复位完成后，单片机从存储器加载初始的堆栈指针，初始的程序计数器，并由程序计数器指定第一条指令，然后开始执行程序。

## 3.2 系统时钟

不同的时钟源都可用来驱动系统时钟（SYSCLK）：

- 48MHz 内部高速 RC 时钟（IHRC）；
- 32KHz 内部低速 RC 时钟（ILRC）；

每个时钟源在不使用时都可以独立的打开或关闭，以使功耗达到最优状态。

该单片机为双时钟系统，包括高速时钟和低速时钟。高速时钟由外部振荡电路和内部 PLL 电路提供；低速时钟则由内部低速 RC 振荡电路（ILRC 32KHz）提供。

### 3.2.1 内部RC时钟源

#### 3.2.1.1 内部高速RC振荡器（IHRC）

内部高速 48MHz RC 振荡器，在普通环境下，其精确度为 $\pm 0.2\%$ 。通过模拟模块控制寄存器（SYS0\_ANBCLTRL）中的 IHRcen 位，可以使 IHRC 自由切换是否使能。

#### 3.2.1.2 内部低速RC振荡器（ILRC）

系统低速时钟源即内置的低速振荡器，采用 RC 振荡电路。低速时钟的输出频率受系统电压和环境温度的影响，通常为 32KHz。

\* 注：只能通过 HW 来切换 ILRC 的状态。



### 3.2.2 系统时钟（SYSCLK）选择

系统复位后，IHRC 作为系统时钟。当直接使用时钟源时，IHRC 仍然正常工作。

若一个时钟源转换到另一个时钟源时，需等待目标时钟源准备好（启动延迟后时钟稳定）。若选择一个未准备好的时钟源，则需等待其准备好之后才会开始转换。

SYS0\_CSST 寄存器的 Ready 位显示已经准备好的时钟源，SYS0\_CLKCFG 寄存器的 SYSCLKST 位则显示的是当前作为系统时钟的时钟源。

### 3.2.3 CLOCK-OUT功能

单片机时钟输出（CLKOUT）性能允许时钟输出到外部 CLKOUT 引脚。此对应 GPIO 的配置寄存器需设定成时钟输出功能。

3 种时钟信号可以作为时钟输出：

- HCLK
- IHRC
- ILRC

由 SYS1\_AHBCLKEN 寄存器的 CLKOUTSEL 位选择控制。

### 3.3 系统控制寄存器 0

基地址: 0x4006 0000

#### 3.3.1 模拟模块控制寄存器 (SYS0\_ANBCTRL)

地址偏移量: 0x00

复位值: 0x0000 0001

\* 注: 若选择 IHRC 作为系统时钟或者准备作为系统时钟, 则 IHRCEN 位不能被清零。

Bit	Name	说明	Attribute	Reset
31:1	Reserved		R	0
0	IHRCEN	内部高速时钟使能位。 0: 禁止内部 48MHz RC 振荡器; 1: 使能内部 48MHz RC 振荡器。	R/W	1

#### 3.3.2 时钟源状态寄存器 (SYS0\_CSST)

地址偏移量: 0x08

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	IHRCRDY	IHRC 时钟准备标志位。IHRCEN 清零后, 6 个 IHRC 后由 HW 清 IHRCRDY。 0: IHRC 没有准备好; 1: IHRC 已经准备好。	R	1

#### 3.3.3 系统时钟配置寄存器 (SYS0\_CLKCFG)

地址偏移量: 0x0C

Bit	Name	说明	Attribute	Reset
31:7	Reserved		R	0
6:4	SYSCLKST[2:0]	系统时钟切换状态。 通过 HW 设置和清零, 以显示当前作为系统时钟的时钟源。 000: IHRC 作为系统时钟; 001: ILRC 作为系统时钟; 其它: 保留。	R	0
3	Reserved		R	0
2:0	SYSCLKSEL[2:0]	系统时钟切换。通过 SW 设置和清零。 000: IHRC; 001: ILRC; 其它: 保留。	R/W	0

#### 3.3.4 AHB时钟预分频寄存器 (SYS0\_AHBCP)

地址偏移量: 0x10

Bit	Name	说明	Attribute	Reset
31:3	Reserved		R	0
2:0	AHBPRES[3:0]	AHB 时钟源预分频值。 000: SYSCLK / 1; 001: SYSCLK / 2; 010: SYSCLK / 4; 011: SYSCLK / 8; 100: SYSCLK / 16; 101: SYSCLK / 32; 110: SYSCLK / 64; 111: SYSCLK / 128; 其它: 保留。	R/W	0

### 3.3.5 系统复位状态寄存器 (SYS0\_RSTST)

地址偏移量: 0x14

该寄存器包含的所有的复位源。

Bit	Name	说明	Attribute	Reset
31:5	Reserved		R	0
4	PORRSTF	POR 复位标志位。发生 POR 复位时，由 HW 设置。 0: Read→ 没有发生 POR 复位; Write→ 将该位清零; 1: 发生 POR 复位。	R/W	1
3	EXTRSTF	外部复位标志位。发生外部复位时，由 HW 设置。 0: Read→ 没有发生外部复位; Write→ 将该位清零; 1: 发生外部复位。	R/W	0
2	LVDRSTF	LVD 复位标志位。发生 LVD 复位时，由 HW 设置。 0: Read→ 没有发生 LVD 复位; Write→ 将该位清零; 1: 发生 LVD 复位。	R/W	0
1	WDTRSTF	WDT 复位标志位。发生 WDT 复位时，由 HW 设置。 0: Read→ 没有发生 WDT 复位; Write→ 将该位清零; 1: 发生 WDT 复位。	R/W	0
0	SWRSTF	软件复位标志位。发生软件复位时，由 HW 设置。 0: Read→ 没有发生软件复位; Write→ 将该位清零; 1: 发生软件复位。	R/W	1

### 3.3.6 LVD控制寄存器 (SYS0\_LVDCTRL)

地址偏移量: 0x18

LVD 寄存器选择 4 个独立的临界值，用于产生一个 LVD 中断或 LVD 复位。

Bit	Name	说明	Attribute	Reset
31:16	Reserved		R	0
15	LV DEN	LVD 使能位。 0: 禁止; 1: 使能。	R/W	0
14	LVDRSTEN	LVD 复位使能位。 0: 禁止→标志; 1: 使能→复位。	R/W	0
13:7	Reserved		R	0
6:5	LV DINTLV L[1:0]	LVD 中断电平。 01: 2.40V; 10: 3.30V; 其它: 保留。	R/W	0
4:3	Reserved		R	0
2:0	LVDRSTLV L[2:0]	LVD 复位电平。 010: 2.40V; 100: 3.30V; 其它: 保留。	R/W	0

### 3.3.7 外部复位引脚控制寄存器 (SYS0\_EXRSTCTRL)

地址偏移量: 0x1C

Bit	Name	说明	Attribute	Reset
31:1	Reserved		R	0
0	RESETDIS	外部复位引脚禁止位。 0: 使能外部复位引脚 (P3.8 作为复位引脚); 1: 禁止 (P3.8 作为 GPIO 引脚)。	R/W	1

### 3.3.8 SWD引脚控制寄存器 (SYS0\_SWDCtrl)

地址偏移量: 0x20

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	SWDDIS	SWD 引脚禁止位。 0: 使能 SWD 引脚 (P3.7 作为 SWDIO 引脚, P3.6 作为 SWCLK 引脚); 1: 禁止 (P3.7 和 P3.6 作为 GPIO 引脚)	R/W	0

### 3.3.9 中断向量映射寄存器 (SYS0\_IVTM)

地址偏移量: 0x24

该寄存器决定 ARM 中断向量映射到用户 ROM 或 SRAM 中。

Bit	Name	Description	Attribute	Reset
31:16	IVTMKEY[15:0]	IVTM 寄存器 Key。读作 0, 除了同时写入 0A5A5H 到 INTMKEY, 将忽略写入该寄存器的动作。	W	0
15:2	Reserved		R	0
1:0	IVTM[1:0]	中断映射选择位。 01: 用户 ROM; 10: SRAM; 00,11: 保留。	R/W	01b

### 3.3.10 噪音检测控制寄存器 (SYS0\_NDTCTRL)

地址偏移量: 0x28

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1	NDT5_IE	NDT0: 禁止 VDD 5V 中断使能位。 0: 禁止; 1: 使能 (由 NDT5V IP 在 IC VDD 5V 检测到噪音会触发 NDT 中断 IRQ0)。	R/W	0
0	Reserved		R	0

### 3.3.11 噪音检测状态寄存器 (SYS0\_NDTSTS)

地址偏移量: 0x2C

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1	NDT5_DET	NDT5V IP 的电源噪音状态。 0: 没有检测到电源噪音; 1: 由 NDT5V IP 检测到电源噪音。*写入 1 到 SYS0_NDTSTS[1]清零。	R/W	0
0	Reserved		R	0

### 3.3.12 Anti-EFT Ability控制寄存器 (SYS0\_ANTIEFT)

地址偏移量: 0x30

该寄存器决定 HW anti-EFT 能力。

Bit	Name	说明	Attribute	Reset
31:3	Reserved		R	0
2:0	AEFT[2:0]	HW anti-EFT 能力。 000: 无; 010: 低; 011: 中; 100: 强。	R/W	000

## 3.4 系统控制寄存器 1

基地址：0x4005 E000

### 3.4.1 AHB时钟使能寄存器（SYS1\_AHBCLKEN）

地址偏移量：0x00

SYS\_AHBCLKEN 寄存器使能 AHB 时钟提供给独立系统和外设模块。

\* 注：

- 1.禁止时钟时，不能通过 SW 读取外设模块的值，该值返回为 0。
- 2.若 CLKOUTSEL 不为 0，则 HW 直接将 GPIO 设置为 CLKOUT 功能。

Bit	Name	说明	Attribute	Reset
31	Reserved		R	0
30:28	CLKOUTPRE[2:0]	时钟输出源分频。 000: 禁止; 001: ILRC; 100: HCLK; 101: IHRC; Others: 保留。	R/W	0
27:25	Reserved		R	0
24	WDTCLKEN	使能 WDT 时钟。 0: 禁止; 1: 使能。	R/W	1
23:22	Reserved		R	0
21	I2C0CLKEN	使能 I2C0 时钟。 0: 禁止; 1: 使能。	R/W	0
20:13	Reserved		R	0
12	SPI0CLKEN	使能 SPI0 时钟。 0: 禁止; 1: 使能。	R/W	0
11:8	Reserved		R	0
7	CT16B1CLKEN	使能 CT16B1 时钟。 0: 禁止; 1: 使能。	R/W	0
6	CT16B0CLKEN	使能 CT16B0 时钟。 0: 禁止; 1: 使能。	R/W	0
5	Reserved		R	0
4	USBCLKEN	使能 USB 时钟。 0: 禁止; 1: 使能。	R/W	0
3	P3CLKEN	使能 P3 时钟。 0: 禁止; 1: 使能。	R/W	1
2	P2CLKEN	使能 P2 时钟。 0: 禁止; 1: 使能。	R/W	1
1	P1CLKEN	使能 P1 时钟。 0: 禁止; 1: 使能。	R/W	1
0	P0CLKEN	使能 P0 时钟。 0: 禁止; 1: 使能。	R/W	1

### 3.4.2 APB时钟预分频寄存器 1 (SYS1\_APBPCP1)

地址偏移量: 0x08

\* 注: 改变预分频值后, 必须通过 SYS1\_PRST 寄存器复位相应的外设。

Bit	Name	Description	Attribute	Reset
31	Reserved		R	0
30:28	CLKOUTPRE[2:0]	时钟源预分频值。 000: 时钟源/1; 001: 时钟源/2; 010: 时钟源/4; 011: 时钟源/8; 100: 时钟源/16; 101: 时钟源/32; 110: 时钟源/64; 111: 时钟源/128。	R/W	0
27:23	Reserved		R	0
22:20	WDTPRE[2:0]	WDT 时钟源预分频值。 000: WDT_PCLK=WDT 时钟源/1; 001: WDT_PCLK=WDT 时钟源/2; 010: WDT_PCLK=WDT 时钟源/4; 011: WDT_PCLK=WDT 时钟源/4; 100: WDT_PCLK=WDT 时钟源/16; 001: WDT_PCLK=WDT 时钟源/32。 Other: 保留。	R/W	0
19:18	Reserved		R	0
17:16	SYSTICKPRE[2:0]	SYS Tick 时钟源预分频值。 00: SYS Tick_PCLK = HCLK / 1; 01: SYS Tick_PCLK = HCLK / 2; 10: SYS Tick_PCLK = HCLK / 4; 11: SYS Tick_PCLK = HCLK / 8。	R/W	0
15:0	Reserved		R	0

# 4 系统操作模式

## 4.1 概述

针对不同的时钟速率以及省电等原因，SN32F260 系列单片机内置 3 种不同的操作模式。这些模式控制振荡器，OP 代码的操作和模拟外设的操作。

- 普通模式；
- 睡眠模式；
- 深度睡眠模式；

## 4.2 普通模式

普通模式下，ARM Cortex-M0 核，存储器，外设都由系统时钟来计时。SYS1\_AHBCLKEN 寄存器控制外设的运行状况。

不同的外设都有独立的外设时钟，同时它们不光有系统时钟，还有自己的时钟分频器，而且这些外设时钟还可以单独禁止或使能。

通过模块的使能位，可以随时单独的控制各种不同的模拟模块（IHRC，Flash 和 LVD）。

## 4.3 低功耗模式

有 2 种模式来降低功耗：睡眠模式和深度睡眠模式。PMU\_CTRL 寄存器控制是否进入不同的低功耗模式。

根据需要，通过改变时钟源，或者改变系统时钟的分频值来改变系统时钟速率，从而可以在应用需求的 CPU 速度和系统功耗之间找到一个平衡点。

程序运行期间，功耗管理可选择关闭芯片上各独立外设的时钟，通过关闭应用中未使用外设的时钟来达到降低功耗的目的。

- \* 注 1：深度睡眠模式下，不支持调试模式。
- \* 注 2：低功耗模式下，将没有输出的引脚设置为合适的模式以减小功耗，强烈建议将这些引脚设置为输入上拉模式。

### 4.3.1 睡眠模式

睡眠模式下，ARM Cortex-M0 核的系统时钟停止工作，指令也暂停执行。

外设功能，若在 SYS1\_AHBCLKEN 寄存器中使能时钟，则睡眠模式下继续工作，并且任何中断可使处理器恢复运行。睡眠模式消除了处理器，内存系统和相关的控制器以及内部总线使用的动态功耗。

模拟模块（IHRC，Flash 和 LVD）的电源状态由模块的使能位决定。

处理器状态和寄存器，外设寄存器和内部 SRAM 值都保持原样，引脚的逻辑电平保持静态。

中断发生时可以将系统从睡眠模式下唤醒。

睡眠模式下，RESET 引脚保持有效状态。

通过下面的步骤，可以进入睡眠模式：

1. 写入 4 到 PMU\_CTRL 寄存器。
2. 执行 ARM Cortex-M0 WFI 指令。

### 4.3.2 深度睡眠模式

在深度睡眠模式下，ARM Cortex-M0 核的系统时钟停止工作，指令也暂停执行。

振荡器的电源状态处于掉电模式时，外设功能的时钟停止工作，时钟源也停止工作。

处理器状态和寄存器，外设寄存器和内部 SRAM 值都保持原样，引脚的逻辑电平保持静态。

所有的 GPIO 引脚都有唤醒功能，用户必须通过 GPIO 寄存器设置 GPIO 引脚的边沿极性，使其可以唤醒 MCU（只有边沿触发才有唤醒功能）。当 GPIO 产生 GPIO 中断事件到 ARM core 时，系统退出深度睡眠模式，此外，必须在 NVIC 中使能中断对应的输入引脚。VDD=3V 时，唤醒时间为 56us。

深度睡眠模式下，RESET 引脚保持有效状态。

通过下面的步骤，可以进入深度睡眠模式：

1. 写入 2 到 PMU\_CTRL 寄存器中。
2. 执行 ARM WFI 指令。

深度睡眠模式的优点在于，它可以使振荡器处于掉电状态，从而更好地降低系统的动态功耗。此外，在深度睡眠模式下可使 Flash 处于掉电状态以降低静态的漏电功耗，但这也会导致唤醒 Flash 存储器需要更长的时间。



## 4.4 系统唤醒

### 4.4.1 概述

系统在睡眠模式下时并不执行程序。唤醒触发信号可以将系统唤醒进入普通模式。唤醒功能内置中断操作，系统被唤醒时触发系统执行中断服务程序。

- 睡眠模式的唤醒触发源为所有中断和 RESET 引脚。
- 深度睡眠的唤醒触发源为 GPIO 中断，CT16B0/1 中断和 RESET 引脚。

### 4.4.2 唤醒时间

系统进入睡眠模式后，由 F/W 使能或禁止高速时钟。若高速时钟停止运行，单片机需要从睡眠模式下唤醒，则单片机需要等待 2048 个外部高速振荡器时钟周期和 32 个内部高速振荡器时钟周期，以使振荡电路进入稳定工作状态，等待的这段时间就称为唤醒时间。唤醒时间结束后，系统才进入到普通模式。

\* 注：若时钟没有停止运行，则系统从睡眠模式唤醒就不需要唤醒时间。

系统在深度睡眠模式下时，高速时钟停止运行。单片机从深度睡眠模式唤醒时，单片机需要等待 10us 的唤醒时间。唤醒时间结束后，系统才进入到普通模式。

高速时钟振荡器的唤醒时间的计算如下：

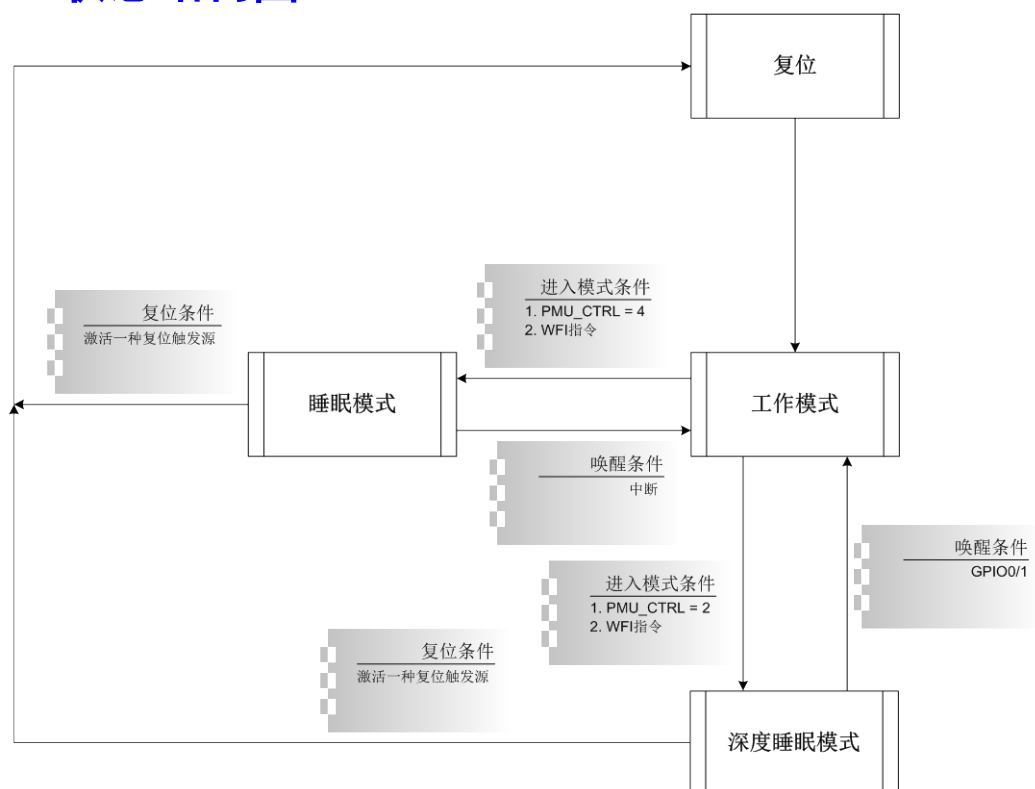
$$\text{总的唤醒时间} = 10\mu\text{s}$$

➤ 例：F<sub>IHR</sub>C=48MHz，唤醒时间计算如下：

$$\text{总的唤醒时间} = 10\mu\text{s} \quad (\text{F}_{\text{IHR}}\text{C} = 48\text{MHz})$$

\* 注：高速时钟的启动时间与 VDD 和振荡器类型有关。

## 4.5 PMU状态结构图



## 4.6 操作模式Comparsion表

工作模式	普通模式	低功耗模式		
		睡眠模式		深度睡眠模式
HCLK	IHRC,ILRC	IHRC	ILRC	-
IHRC	使能 HW	IHRCEN	禁止 FW	禁止 HW
ILRC	使能 HW	使能 HW	使能 HW	禁止 HW
Cortex-M0	运行	停止	停止	停止
Flash ROM	使能/备用	备用	备用	使能/备用
Data RAM	使能/备用	备用	备用	备用
3 级 LVD	LV DEN	LV DEN		LV DEN
USB	USBEN	USBEN		禁止
外设	外设的使能位			禁止 HCLK
IO 状态	输出低电平	输出低电平		输出低电平
唤醒源	N/A	所有中断，复位引脚		GPIO 中断，复位引脚

## 4.7 PMU寄存器

基地址：0x4003 2000

### 4.7.1 电源控制寄存器（PMU\_CTRL）

地址偏移量：0x40

电源控制寄存器可选择控制 ARM Cortex-M0 进入不同的掉电模式（睡眠模式或深度睡眠模式），并且提供独立的标志位来指示睡眠模式或深度睡眠模式。

\* 注：在深度掉电模式下，只要 VDD 引脚仍然接有电源，PMU\_CTRL 寄存器就会保留数据；只有在芯片的所有电源都被断开后，此寄存器才会复位。

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2:0	MODE[2:0]	低功耗模式选择控制位。 010: WFI 指令使 MCU 进入深度睡眠模式； 100: WFI 指令使 MCU 进入睡眠模式； 其它：禁止。	R/W	0

# 5

## GPIO端口

### 5.1 概述

GPIO 端口可以通过 SW 来设置为输入或输出模式。

- 每个独立的引脚可以作为外部中断输入引脚使用。
- 可以通过单一的上升沿或者下降沿，或者双边沿触发来触发中断。
- 通过程序控制独立的中断级别，内部上拉电阻。
- 所有的 GPIO 引脚可作为输入引脚，或默认为悬浮状态。

### 5.2 GPIO模式

GPIO<sub>n</sub>\_CFG (n = 0、1、2、3) 寄存器的 MODE 位配置每个引脚的上拉电阻，或者选择 inactive 模式或者 inactive 模式且禁止施密特触发。

内置电阻的配上可以是：使能上拉电阻；无上拉/下拉电阻，使能施密特触发（默认模式）；或者无上拉/下拉电阻，但禁止施密特触发

## 5.3 GPIO寄存器

基地址：0x4004 4000 (GPIO 0)

0x4004 6000 (GPIO 1)

0x4004 8000 (GPIO 2)

0x4004 A000 (GPIO 3)

### 5.3.1 GPIO Pn数据寄存器 (GPIO<sub>n</sub>\_DATA) (n=0,1,2,3)

地址偏移量：0x00

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	DATA[15:0]	输入数据（读）或输出数据（写）。	R/W	0

### 5.3.2 GPIO Pn模式寄存器 (GPIO<sub>n</sub>\_MODE) (n=0,1,2,3)

地址偏移量：0x04

\* 注：使能特殊功能（外设）时，HW 会直接切换 I/O 模式，而不通过 GPIO<sub>n</sub>\_MODE 寄存器。

Bit	Name	Description	Attribute	Reset
31:16	CURRENT[15:0]	驱动/灌电流当前值选择位（x=0~15）。 0：10mA；1：20mA。	R/W	0
15:0	MODE[15:0]	选择引脚 x 作为输入或输出引脚（x = 0~15） 0：输入引脚； 1：输出引脚。	R/W	0

### 5.3.3 GPIO Pn配置寄存器 (GPIO<sub>n</sub>\_CFG) (n=0,1,2,3)

地址偏移量：0x08

复位值：n = 0→0xAAAAAAAA

n = 1→0x00000AAA

n = 2→0x002AAAAA

n = 3→0x0002AAAA

\* 注：使能特殊功能（外设）时，HW 会直接切换 I/O 模式，而不通过 GPIO<sub>n</sub>\_MODE 寄存器。

Bit	Name	Description	Attribute	Reset
31:30	CFG15[1:0]	Pn.15 的配置位。 00：使能上拉电阻；01：保留 10：inactive 模式（禁止上拉电阻，使能施密特触发）； 11：inactive 模式（禁止上拉电阻，禁止施密特触发，数据寄存器为低电平）。	R/W	10b
29:28	CFG14[1:0]	Pn.14 的配置位。 00：使能上拉电阻；01：保留 10：inactive 模式（禁止上拉电阻，使能施密特触发）； 11：inactive 模式（禁止上拉电阻，禁止施密特触发，数据寄存器为低电平）。	R/W	10b
27:26	CFG13[1:0]	Pn.13 的配置位。 00：使能上拉电阻；01：保留 10：inactive 模式（禁止上拉电阻，使能施密特触发）； 11：inactive 模式（禁止上拉电阻，禁止施密特触发，数据寄存器为低电平）。	R/W	10b
25:24	CFG13[1:0]	Pn.13 的配置位。	R/W	10b

		00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。		
23:22	CFG11[1:0]	Pn.11 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。。	R/W	10b
21:20	CFG10[1:0]	Pn.10 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。	R/W	10b
19:18	CFG9[1:0]	Pn.9 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。	R/W	10b
17:16	CFG8[1:0]	Pn.8 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。	R/W	10b
15:14	CFG7[1:0]	Pn.7 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。	R/W	10b
13:12	CFG6[1:0]	Pn.6 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。	R/W	10b
11:10	CFG5[1:0]	Pn.5 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。	R/W	10b
9:8	CFG4[1:0]	Pn.4 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。	R/W	10b
7:6	CFG3[1:0]	Pn.3 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。	R/W	10b
5:4	CFG2[1:0]	Pn.2 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。	R/W	10b
3:2	CFG1[1:0]	Pn.1 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。	R/W	10b
1:0	CFG0[1:0]	Pn.0 的配置位。 00: 使能上拉电阻; 01: 保留 10: inactive 模式 (禁止上拉电阻, 使能施密特触发); 11: inactive 模式 (禁止上拉电阻, 禁止施密特触发, 数据寄存器为低电平)。	R/W	10b

### 5.3.4 GPIO Pn中断SENSE寄存器 (GPIO<sub>n</sub>\_IS) (n=0,1,2,3)

地址偏移量: 0x0C

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IS[15:0]	选择 Pn.x 是由电平触发还是边沿触发 (x = 0~15)。 0: 边沿触发; 1: 事件触发。	R/W	0

### 5.3.5 GPIO Pn中断双边沿SENSE寄存器 (GPIO<sub>n</sub>\_IBS) (n=0,1,2,3)

地址偏移量: 0x10

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IBS[15:0]	选择是否由双边沿触发 Pn.x 的中断 (x = 0~15)。 0: 由寄存器 GPIO <sub>n</sub> _IEV 控制 Pn.x 的中断; 1: 由双边沿触发 Pn.x 的中断。	R/W	0

### 5.3.6 GPIO Pn中断事件寄存器 (GPIO<sub>n</sub>\_IEV) (n=0,1,2,3)

地址偏移量: 0x14

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IEV[15:0]	选择 Pn.x 是在上升沿还是下降沿触发中断 (x = 0~15)。 0: 取决于寄存器 GPIO <sub>n</sub> _IS 的设置, 上升沿或者 Pn.x 为高电平触发中断; 1: 取决于寄存器 GPIO <sub>n</sub> _IS 的设置, 下降沿或者 Pn.x 为低电平触发中断。	R/W	0

### 5.3.7 GPIO Pn中断使能寄存器 (GPIO<sub>n</sub>\_IE) (n=0,1,2,3)

地址偏移量: 0x18

将 GPIO<sub>n</sub>\_IE 寄存器的位置 1 时, 允许相对应的引脚触发各自的中断; 反之将寄存器的位清零则禁止各自的中断。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IE[15:0]	Pn.x 中断使能位 (x = 0~15)。 0: 禁止; 1: 使能。	R/W	0

### 5.3.8 GPIO Pn中断源状态寄存器 (GPIO<sub>n</sub>\_RIS) (n=0,1,2,3)

地址偏移量: 0x1C

该寄存器显示 GPIO 控制中断源的状态, 若 GPIO<sub>n</sub>\_IE 寄存器的位置 1, 则发送 GPIO 中断到中断控制器中。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IF[15:0]	GPIO 中断源标志位 (x = 0~15)。 0: 无中断; 1: 有中断发生。	R	0

### 5.3.9 GPIO Pn中断清零寄存器（GPIO<sub>n</sub>\_IC）（n=0,1,2,3）

地址偏移量：0x20

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	IC[15:0]	Pn.x 中断标志清零使能位（x = 0~5）。 0：没有影响；1：清除 Pn.x 中断标志位。	W	0

### 5.3.10 GPIO Pn位设置操作寄存器（GPIO<sub>n</sub>\_BSET）（n=0,1,2,3）

地址偏移量：0x24

为了让 SW 设置 GPIO 位而不影响到其它引脚的写操作，若该寄存器的相应位置 1，则 GPIO 位也置 1。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	BSET[15:0]	位设置使能位（x = 0~15）。 0：不影响 Pn.x； 1：Pn.x 位置 1。	W	0

### 5.3.11 GPIO Pn位清零操作寄存器（GPIO<sub>n</sub>\_BCLR）（n=0,1,2,3）

地址偏移量：0x28

为了让 SW 将 GPIO 位清零而不影响到其它引脚的写操作，若该寄存器的相应位置 1，则 GPIO 位被清零。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	BCLR[15:0]	位清零使能位（x = 0~15）。 0：不影响 Pn.x； 1：Pn.x 位清零。	W	0



# 6 16 位带捕获功能定时器

## 6.1 概述

16 位计数/定时器用于对外设时钟（PCLK）信号或外部提供的时钟信号进行计数。根据 4 个匹配寄存器中指定的定时时间，可以使定时器有选择地产生中断请求或完成其它操作。每个计数/定时器都有一路输入信号捕获通道，当输入信号发生变化时，便可将此时间值记录下来，可选择捕获功能触发中断。

PWM 模式下，高达 23 个匹配和 1 个全局寄存器可以用于提供单边沿信号的 PWM 输出到匹配输出引脚上。

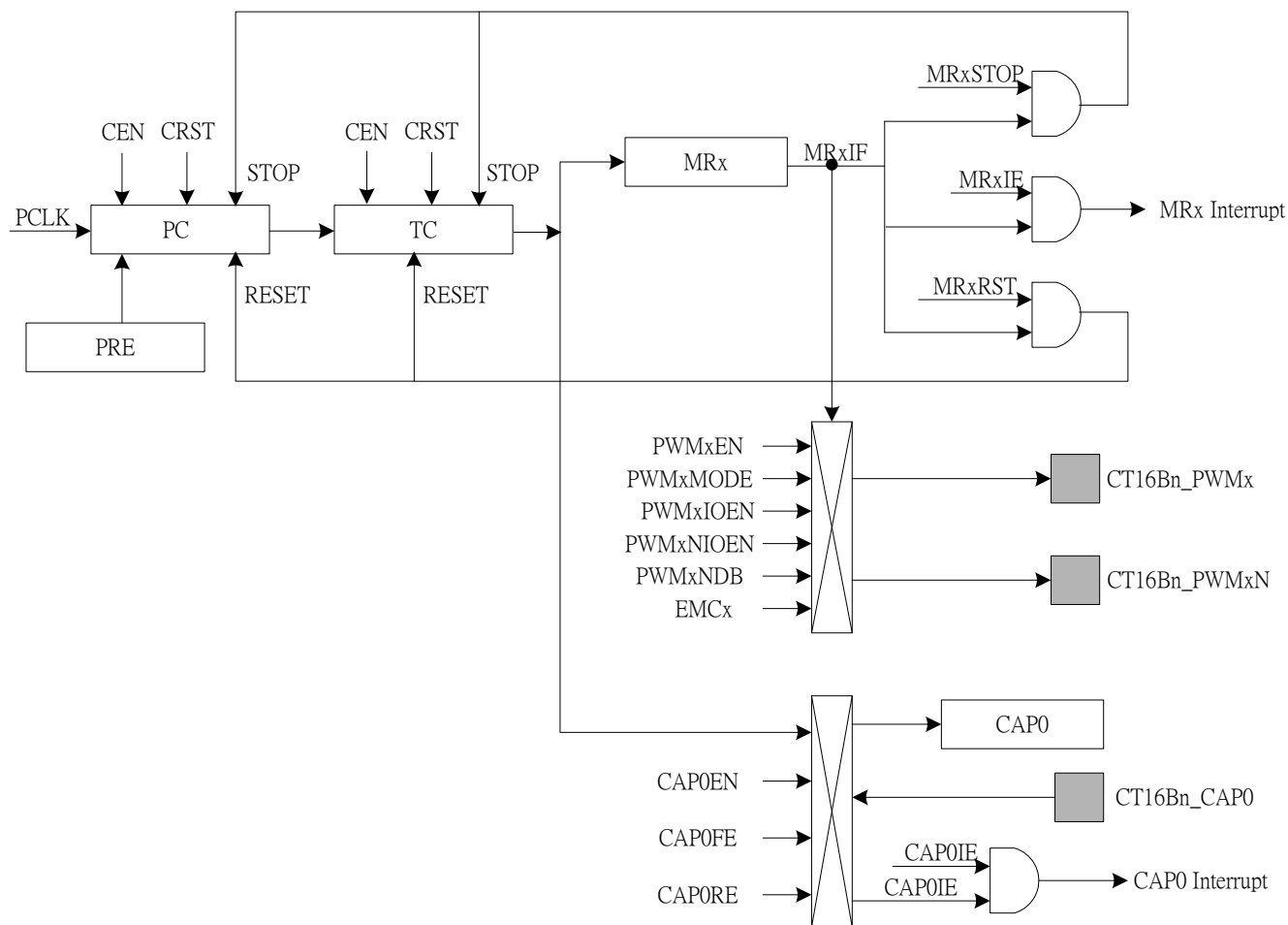
## 6.2 特性

- 2 个 16 位计数/定时器。
- 计数器和定时器操作。
- 2 路 16 位捕获通道，当有输入信号发生改变时，可以迅速记录下时间。每一捕获事件都可以选择是否产生中断。
- 可以设定定时器时间值在指定的捕获事件发生时被清零。这一特性可以应用于简单的脉宽测量，通过在输入脉冲的起始边沿清除定时器，在输入脉冲的尾端边沿便可取得此脉冲宽度的时间值。
- 4 个 16 位匹配寄存器有如下功能：
  - 连续计数，可选择在计数到匹配数值时是否产生中断。
  - 在计数到匹配数据时停止计数，可选择是否产生中断。
  - 在计数到匹配数据时复位定时器，可选择是否产生中断。
- 多达 22 路（CT16B1）PWM 输出，每路 PWM 输出都和某一匹配寄存器一一对应，具体功能如下：
  - 在计数到匹配数据时，输出低。
  - 在计数到匹配数据时，输出高。
  - 在计数到匹配数据时，反转电平。
  - 在计数到匹配数据时，保持状态。

## 6.3 引脚说明

Pin Name	Type	Description	GPIO Configuration
CT16Bn_CAP0	I	输入 0 的捕获通道	取决于 GPIO <sub>n</sub> _CFG
CT16Bn_PWMx	O	Match/PWM 的输出通道 x。	

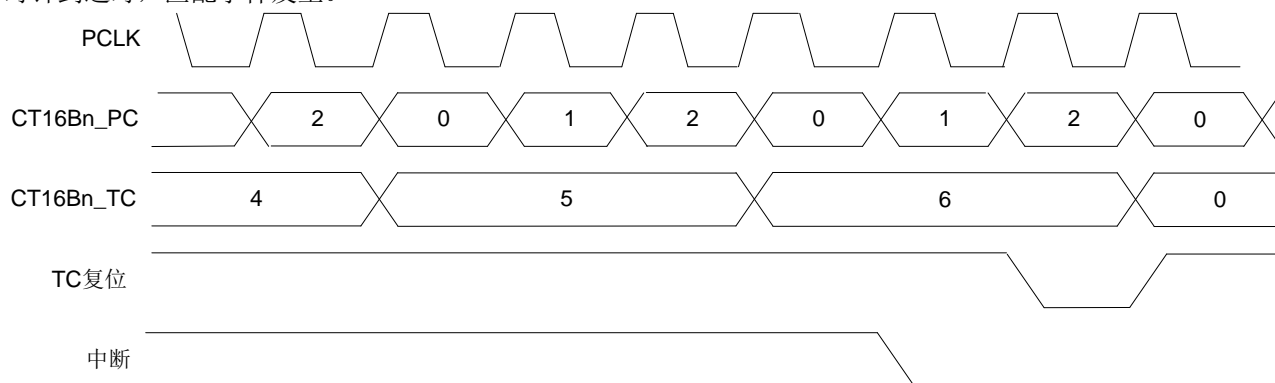
## 6.4 框图



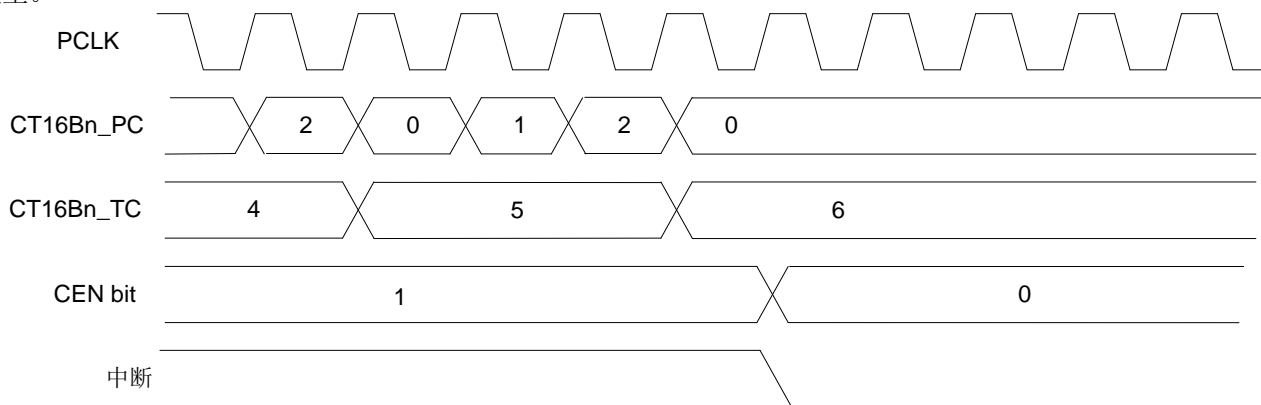
## 6.5 定时器操作

### 6.5.1 边沿对齐向上计数模式

下图显示一被配置过的定时器在边沿对齐向上计数模式时复位计数并产生中断请求。CT16Bn\_PRE 寄存器设置为 2，CT16Bn\_MRx 设置为 6。匹配事件发生后，在定时器周期结束时定时器复位。当定时器计数值达到匹配值后，在它的下一个时钟到达时，匹配事件发生。



下图显示定时器被配置成在边沿对齐向上计数模式时停止计数，并产生中断。CT16Bn\_PRE 寄存器设置为 2，CT16Bn\_MRx 设置为 6。定时器达到匹配值后的下一个时钟内，CT16Bn\_TMRCTRL 寄存器的 CEN 位清零，中断显示匹配事件发生。



## 6.6 PWM

### 6.6.1 PWM模式 1

在向上计数时，若  $TC < MR_n$ ，则  $PWM_n$  为 0。

边沿对齐向上计数模式操作流程如下：

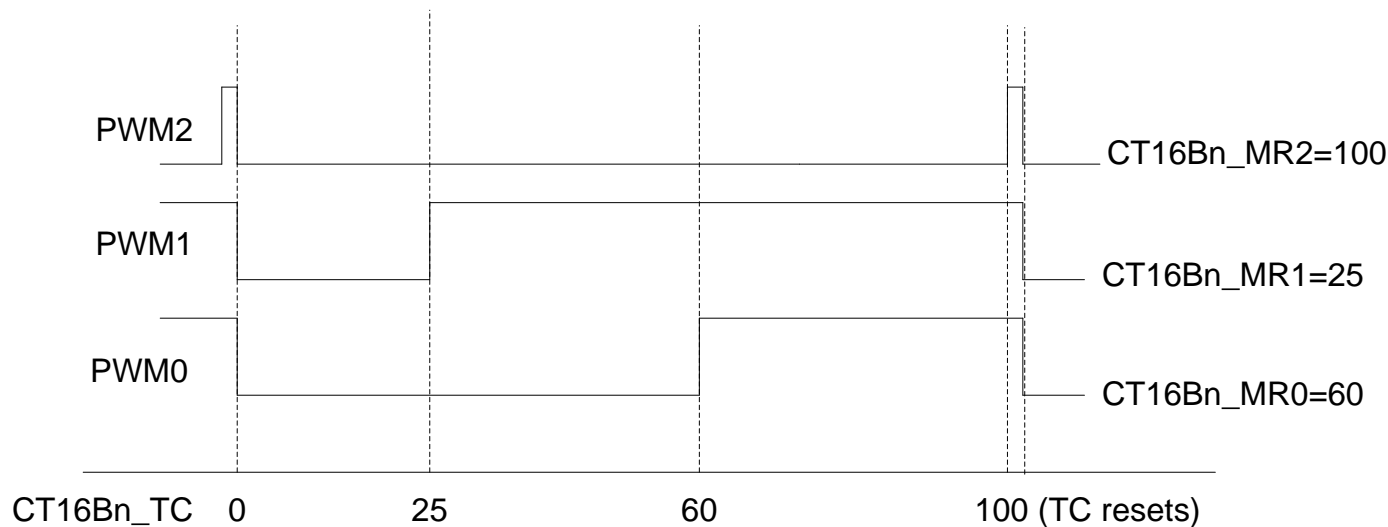
1、在每个 PWM 周期的起始端（定时器设置为 0），所有信号边沿被 PWM 控制输出低电平，除非 CT16Bn\_MR0~3 寄存器的匹配值为 0。

2、到达匹配值后，每个 PWM 输出高电平；若没有达到匹配值，则继续输出低电平。

3、若写入 CT16Bn\_MR0~3 寄存器的匹配值大于 PWM 周期长度，且 PWM 信号已经是高电平，则在下一个 PWM 周期开始时清除 PWM 信号。

4、若匹配寄存器的值和定时器的复位值（PWM 周期长度）相同，则 PWM 在下一个时钟 tick 输出低电平。因此，PWM 将一直输出由时钟 tick 宽度决定的正极脉冲，其周期取决于 PWM 周期的长度。

5、若匹配寄存器设置为 0，PWM 首先输出高电平，定时器返回到 0，PWM 会一直保持高电平。



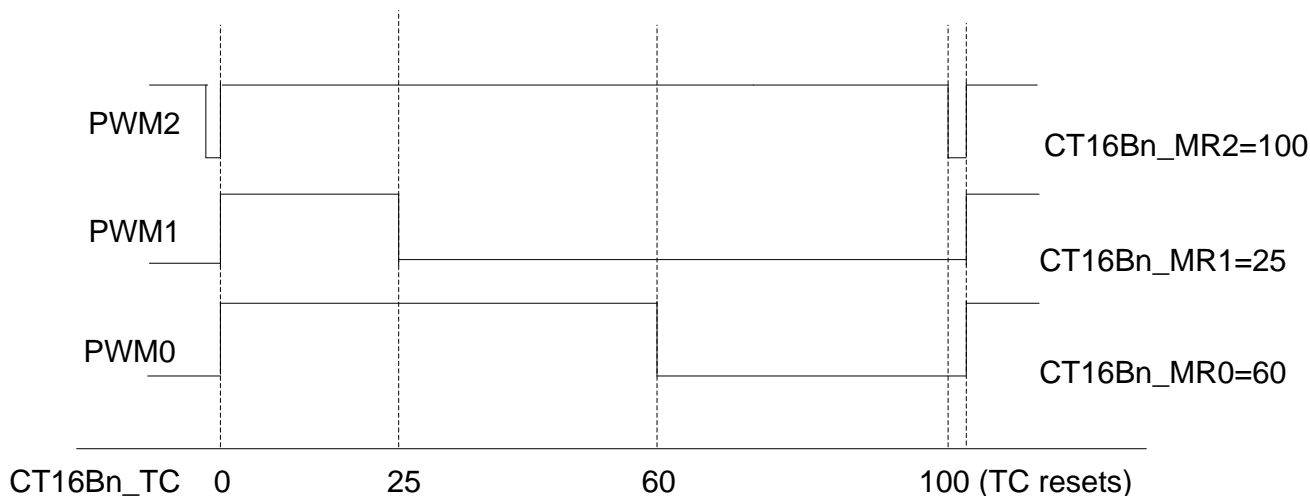
\* 注：当选择的匹配输出作为 PWM 输出执行时，除了匹配寄存器需要设置 PWM 周期长度外，CT16Bn\_MCTRL 寄存器的定时器复位位（MRnRST）和定时器停止位（MRnSTOP）都必须为 0。当相应匹配寄存器的值和定时器的值相匹配时，设置 MRnR 位为 1 以使能定时器复位。

## 6.6.2 PWM模式 2

在向上计数时，若  $TC < MR_n$ ，则  $PWM_n$  为 1。

边沿对齐向上计数模式操作流程如下：

- 1、在每个 PWM 周期的起始端（定时器设置为 0），所有信号边沿被 PWM 控制输出高电平，除非 CT16Bn\_MR0~3 寄存器的匹配值为 0。
- 2、到达匹配值后，每个 PWM 输出低电平；若没有达到匹配值，则继续输出高电平。
- 3、若写入 CT16Bn\_MR0~3 寄存器的匹配值大于 PWM 周期长度，且 PWM 信号已经是低电平，则在下个 PWM 周期开始时输出高电平。
- 4、若匹配寄存器的值和定时器的复位值（PWM 周期长度）相同，则 PWM 在下个时钟 tick 输出高电平。因此，PWM 将一直输出由时钟 tick 宽度组成的低电平脉冲，其周期取决于 PWM 周期的长度。
- 5、若匹配寄存器设置为 0，PWM 首先输出低电平，定时器返回到 0，PWM 会一直保持低电平。



\* 注：当选择的匹配输出作为 PWM 输出执行时，除了匹配寄存器需要设置 PWM 周期长度外，CT16Bn\_MCTRL 寄存器的定时器复位位（MRnRST）和定时器停止位（MRnSTOP）都必须为 0。当相应匹配寄存器的值和定时器的值相匹配时，设置 MRnR 位为 1 以使能定时器复位。

## 6.7 CT16Bn寄存器

基地址: 0x4000 0000 (CT16B0)  
0x4000 2000 (CT16B1)

### 6.7.1 CT16Bn定时器控制寄存器 (CT16Bn\_TMRCTRL) (n=0,1)

地址偏移量: 0x00

\* 注: 为准确初始化 TC 和 PC, SW 通过设置 CRST 为 1 来复位 TC 和 PC, 并通过设置 CRST 为 1 了使能计数器。

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1	CRST	计数器复位位。 0: 禁止计数器复位; 1: 在 PCLK 的下个正极边沿, 定时/计数器同步复位, 复位完成后由 HW 将该位清零。	R/W	0
0	CEN	计数器使能位。 0: 禁止; 1: 使能定时器/计数器计数。一个最后设置 CEN 位。*通常在边沿对齐向上计数模式。	R/W	0

### 6.7.2 CT16Bn定时器计数器寄存器 (CT16Bn\_TC) (n=0,1)

地址偏移量: 0x04

预分频计数器到达终端值时, 16 位定时计数器递增计数, 除非计数器在到达上限值之前复位, 否则 TC 一直计数到 0x0000FFFF, 然后再返回 0x00000000。这并不会引起中断, 但有需要的时候, 可以由匹配寄存器来检测是否溢出。

Bit	Name	Description	Attribute	Reset
31:0	Reserved		R	0
15:0	TC[15:0]	定时器计数器。	R/W	0

### 6.7.3 CT16Bn前置分频寄存器 (CT16Bn\_PRE) (n=0,1)

地址偏移量: 0x08

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PR[7:0]	前置分频最大值。	R/W	0

### 6.7.4 CT16Bn前置分频计数器寄存器 (CT16Bn\_PC) (n=0,1, 2)

地址偏移量: 0x0C

在应用于定时器计数器之前, 8 位预分频计数器通过部分恒定值控制 PCLK 的分频。它可以控制定时器的分辨率和最大值之间的关系。预分频计数器在每个 PCLK 递增计数, 在达到存储在预分频寄存器的值后, 定时器计数器继续计数, 而预分频计数器在下个 PCLK 复位。它促使下列情况出现: PR=0 时, TC 在每个 PCLK 递增计数; PR=1 是, 则在每 2 个 PCLK 递增计数。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	PC[7:0]	预分频计数器	R/W	0

## 6.7.5 CT16Bn计数控制寄存器 (CT16Bn\_CNTCTRL) (n=0)

地址偏移量: 0x10

该寄存器用于选择定时器模式或者计数器模式，计数器模式下选择用于计数的引脚和边沿。

选择计数器模式作为操作模式时，CAP 输入（由 CIS 位选择）在 PCLK 时钟的每个上升沿采样。比较两个连续的 CAP 输入采样信号后，在选择的 CAP 输入电平上，下列 4 个事件中的其中一件是经过验证的：上升沿，下降沿，边沿或者 CAP 输入电平无变换。只有确认过的事件发生，而且该事件符合该寄存器的 CTM 位选择的事件，定时器计数器寄存器才会递增计数。

为了使外部提供给计数器的时钟信号可以得到有效处理，计数器对提供的外部时钟信号会有所限制。因为两个连续的 PCLK 时钟的上升沿才能确定 CAP 输入引脚上输入信号的一个跳变边沿，所以 CAP 输入信号的频率不能超过 PCLK 时钟频率的 1/2，即通过 CAP 输入的高/低电平的持续时间不能短于  $1/(2 \times \text{PCLK})$ 。

\* 注：在 CNTCTRL 寄存器下选择计数器模式，捕获控制寄存器 (CAPCTRL) 的 bit2~0 必须由程序设置为 0x00。

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3:2	CIS[1:0]	计数输入选择。在计数器模式 (CTM[1:0]不是 00) 下，该位选择 CAP 引脚对时钟进行采样。 00: CT16Bn_CAP0; 其它: 保留。	R/W	0
1:0	CTM[1:0]	计数器/定时器模式。此字段选择 PCLK 的上升沿将 PC 清零并且定时器计数器递增计数。 00: 定时器模式，PCLK 的每个上升沿; 01: 计数器模式，TC 在 CIS 位选择的 CAP0 输入的上升沿递增计数; 10: 计数器模式，TC 在 CIS 位选择的 CAP0 输入的下降沿递增计数; 11: 计数器模式，TC 在 CIS 位选择的 CAP0 输入的上升/下降沿递增计数。	R/W	0

## 6.7.6 CT16Bn匹配控制寄存器 (CT16Bn\_MCTRL) (n=0)

地址偏移量: 0x14

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	MR0STOP	停止 MR0: 若 MR0 和 TC 匹配，TC 停止工作，CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
1	MR0RST	MR0 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
0	MR0IE	MR0 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0

**6.7.7 CT16Bn匹配控制寄存器（CT16Bn\_MCTRL）（n=1）**

地址偏移量：0x14

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R	0
29	MR9STOP	停止 MR9：若 MR9 和 TC 匹配，TC 停止工作，CEN 位清零。 0：禁止； 1：使能。	R/W	0
28	MR9RST	MR9 和 TC 匹配时复位 TC。 0：禁止； 1：使能。	R/W	0
27	MR9IE	MR9 和 TC 的值匹配时产生中断。 0：禁止； 1：使能。	R/W	0
26	MR8STOP	停止 MR8：若 MR8 和 TC 匹配，TC 停止工作，CEN 位清零。 0：禁止； 1：使能。	R/W	0
25	MR8RST	MR8 和 TC 匹配时复位 TC。 0：禁止； 1：使能。	R/W	0
24	MR8IE	MR8 和 TC 的值匹配时产生中断。 0：禁止； 1：使能。	R/W	0
23	MR7STOP	停止 MR7：若 MR7 和 TC 匹配，TC 停止工作，CEN 位清零。 0：禁止； 1：使能。	R/W	0
22	MR7RST	MR7 和 TC 匹配时复位 TC。 0：禁止； 1：使能。	R/W	0
21	MR7IE	MR7 和 TC 的值匹配时产生中断。 0：禁止； 1：使能。	R/W	0
20	MR6STOP	停止 MR6：若 MR6 和 TC 匹配，TC 停止工作，CEN 位清零。 0：禁止； 1：使能。	R/W	0
19	MR6RST	MR6 和 TC 匹配时复位 TC。 0：禁止； 1：使能。	R/W	0
18	MR6IE	MR6 和 TC 的值匹配时产生中断。 0：禁止； 1：使能。	R/W	0
17	MR5STOP	停止 MR5：若 MR5 和 TC 匹配，TC 停止工作，CEN 位清零。 0：禁止； 1：使能。	R/W	0
16	MR5RST	MR5 和 TC 匹配时复位 TC。 0：禁止； 1：使能。	R/W	0
15	MR5IE	MR5 和 TC 的值匹配时产生中断。 0：禁止； 1：使能。	R/W	0
14	MR4STOP	停止 MR4：若 MR4 和 TC 匹配，TC 停止工作，CEN 位清零。 0：禁止； 1：使能。	R/W	0
13	MR4RST	MR4 和 TC 匹配时复位 TC。 0：禁止； 1：使能。	R/W	0
12	MR4IE	MR4 和 TC 的值匹配时产生中断。 0：禁止； 1：使能。	R/W	0
11	MR3STOP	停止 MR3：若 MR3 和 TC 匹配，TC 停止工作，CEN 位清零。	R/W	0



		0: 禁止; 1: 使能。		
10	MR3RST	MR3 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
9	MR3IE	MR3 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
8	MR2STOP	停止 MR2: 若 MR2 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
7	MR2RST	MR2 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
6	MR2IE	MR2 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
5	MR1STOP	停止 MR1: 若 MR1 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
4	MR1RST	MR1 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
3	MR1IE	MR1 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
2	MR0STOP	停止 MR0: 若 MR0 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
1	MR0RST	MR0 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
0	MR0IE	MR0 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0

## 6.7.8 CT16Bn匹配控制寄存器 2 (CT16Bn\_MCTRL2) (n=1)

地址偏移量: 0x18

Bit	Name	Description	Attribute	Reset
31:30	Reserved		R	0
29	MR19STOP	停止 MR19: 若 MR19 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
28	MR19RST	MR19 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
27	MR19IE	MR19 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
26	MR18STOP	停止 MR18: 若 MR18 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
25	MR18RST	MR18 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
24	MR18IE	MR18 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
23	MR17STOP	停止 MR17: 若 MR17 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
22	MR17RST	MR17 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
21	MR17IE	MR17 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
20	MR16STOP	停止 MR16: 若 MR16 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
19	MR16RST	MR16 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
18	MR16IE	MR16 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
17	MR15STOP	停止 MR15: 若 MR15 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
16	MR15RST	MR15 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
15	MR15IE	MR15 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
14	MR14STOP	停止 MR14: 若 MR14 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
13	MR14RST	MR14 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
12	MR14IE	MR14 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
11	MR13STOP	停止 MR13: 若 MR13 和 TC 匹配, TC 停止工作, CEN 位清零。	R/W	0

		0: 禁止; 1: 使能。		
10	MR13RST	MR13 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
9	MR13IE	MR13 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
8	MR12STOP	停止 MR12: 若 MR12 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
7	MR12RST	MR12 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
6	MR12IE	MR12 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
5	MR11STOP	停止 MR11: 若 MR11 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
4	MR11RST	MR11 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
3	MR11IE	MR11 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
2	MR10STOP	停止 MR10: 若 MR10 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
1	MR10RST	MR10 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
0	MR10IE	MR10 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0

### 6.7.9 CT16Bn匹配控制寄存器 3 (CT16Bn\_MCTRL3) (n=1)

地址偏移量: 0x1C

Bit	Name	Description	Attribute	Reset
31:12	Reserved		R	0
11	MR23STOP	停止 MR23: 若 MR23 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
10	MR23RST	MR23 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
9	MR23IE	MR23 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
8	MR22STOP	停止 MR22: 若 MR22 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
7	MR22RST	MR22 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
6	MR22IE	MR22 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
5	MR21STOP	停止 MR21: 若 MR21 和 TC 匹配, TC 停止工作, CEN 位清零。 0: 禁止; 1: 使能。	R/W	0
4	MR21RST	MR21 和 TC 匹配时复位 TC。 0: 禁止; 1: 使能。	R/W	0
3	MR21IE	MR21 和 TC 的值匹配时产生中断。 0: 禁止; 1: 使能。	R/W	0
2:0	Reserved		R	0

### 6.7.10 CT16Bn匹配寄存器 0 (CT16Bn\_MR) (n=0)

地址偏移量: 0x20

匹配寄存器的值不断地与定时器计数器 (TC) 的值进行比较, 当 2 个值相等时, 自动触发动作。该动作可能是产生中断, 复位 TC 或者停止定时器, 通过设置 CT16Bn\_MCTRL 寄存器控制。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	MR[15:0]	TC 匹配值。	R/W	0

### 6.7.11 CT16Bn匹配寄存器 0~19,21~23 (CT16Bn\_MR0~19,21~23) (n=1)

MR 0~19

地址偏移量: 0x20, 0x24, 0x28, 0x2C, 0x30, 0x34, 0x38, 0x3C, 0x40, 0x44, 0x48, 0x4C, 0x50, 0x54, 0x58, 0x5C, 0x60, 0x64, 0x68, 0x6C;

MR21~23

地址偏移量: 0x74, 0x78, 0x7C

匹配寄存器的值不断地与定时器计数器 (TC) 的值进行比较, 当 2 个值相等时, 自动触发动作。该动作可能是产生中断, 复位 TC 或者停止定时器, 通过设置 CT16Bn\_MCTRL 寄存器控制。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	MR[15:0]	TC 匹配值。	R/W	0

### 6.7.12 CT16Bn捕获控制寄存器 (CT16Bn\_CAPCTRL) (n=0)

地址偏移量: 0x80

发生捕获事件时, 捕获寄存器是否将值装入计时器/定时器, 或者是否由捕获事件产生中断, 都通过捕获控制寄存器来控制。同时设置上升位和下降位为有效配置, 这样导致上升/下降沿发生捕获事件。

\* 注: 若 CNTCTRL 寄存器选择计数器模式, 必须由程序设置 CAPCTRL[2:0]为 0x00。

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	CAP0EN	捕获 0 功能使能位。 0: 禁止; 1: 使能。HW 直接切换 I/O 配置。	R/W	0
2	CAP0IE	CT16Bn_CAP0 事件中断: CT16Bn_CAP0 事件产生中断。 0: 禁止; 1: 使能。	R/W	0
1	CAP0FE	CT16Bn_CAP0 下降沿捕获: 当 CT16Bn_CAP0 检测到输入信号按照先 1 后 0 的顺序变化时, CAP0 保存此时 TC 数值。 0: 禁止; 1: 使能。	R/W	0
0	CAP0RE	CT16Bn_CAP0 上升沿捕获: 当 CT16Bn_CAP0 检测到输入信号按照先 0 后 1 的顺序变化时, CAP0 保存此时 TC 数值。 0: 禁止; 1: 使能。	R/W	0

### 6.7.13 CT16Bn捕获 0 寄存器 (CT16Bn\_CAP0) (n=0)

地址偏移量: 0x84

每个捕获寄存器都与一个设备引脚相对应, 当此设备引脚上有指定的事件发生时, 计数器/定时器数值便可以保存在对应的捕获寄存器中。捕获控制寄存器的设置决定是否使能捕获功能, 以及捕获事件是否在相关联的引脚的上升沿、下降沿或双边沿发生。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	CAP0[15:0]	定时器计数器捕获值。	R	0

### 6.7.14 CT16Bn外部匹配寄存器（CT16Bn\_EM）（n=1）

地址偏移量：0x88

外部匹配寄存器提供 CT16Bn\_PWM[22:0]的控制和状态。若匹配输出配置为 PWM 输出，外部匹配寄存器的功能由 PWM 规则决定。

Bit	Name	Description	Attribute	Reset
31:23	Reserved		R	0
22	EM22	TC 和 MR22 相等时, 该位根据 EMC22 位进行动作, 并驱动 CT16Bn_PWM22 输出状态。	R/W	0
21	EM21	TC 和 MR21 相等时, 该位根据 EMC21 位进行动作, 并驱动 CT16Bn_PWM21 输出状态。	R/W	0
20	Reserved		R	0
19	EM19	TC 和 MR19 相等时, 该位根据 EMC19 位进行动作, 并驱动 CT16Bn_PWM19 输出状态。	R/W	0
18	EM18	TC 和 MR18 相等时, 该位根据 EMC18 位进行动作, 并驱动 CT16Bn_PWM18 输出状态。	R/W	0
17	EM19	TC 和 MR17 相等时, 该位根据 EMC17 位进行动作, 并驱动 CT16Bn_PWM17 输出状态。	R/W	0
16	EM18	TC 和 MR16 相等时, 该位根据 EMC16 位进行动作, 并驱动 CT16Bn_PWM16 输出状态。	R/W	0
15	EM19	TC 和 MR15 相等时, 该位根据 EMC15 位进行动作, 并驱动 CT16Bn_PWM15 输出状态。	R/W	0
14	EM18	TC 和 MR14 相等时, 该位根据 EMC14 位进行动作, 并驱动 CT16Bn_PWM14 输出状态。	R/W	0
13	EM19	TC 和 MR13 相等时, 该位根据 EMC13 位进行动作, 并驱动 CT16Bn_PWM13 输出状态。	R/W	0
12	EM18	TC 和 MR12 相等时, 该位根据 EMC12 位进行动作, 并驱动 CT16Bn_PWM12 输出状态。	R/W	0
11	EM11	TC 和 MR11 相等时, 该位根据 EMC11 位进行动作, 并驱动 CT16Bn_PWM11 输出状态。	R/W	0
10	EM10	TC 和 MR10 相等时, 该位根据 EMC10 位进行动作, 并驱动 CT16Bn_PWM10 输出状态。	R/W	0
9	EM9	TC 和 MR9 相等时, 该位根据 EMC9 位进行动作, 并驱动 CT16Bn_PWM9 输出状态。	R/W	0
8	EM8	TC 和 MR8 相等时, 该位根据 EMC8 位进行动作, 并驱动 CT16Bn_PWM8 输出状态。	R/W	0
7	EM7	TC 和 MR7 相等时, 该位根据 EMC7 位进行动作, 并驱动 CT16Bn_PWM7 输出状态。	R/W	0
6	EM6	TC 和 MR6 相等时, 该位根据 EMC6 位进行动作, 并驱动 CT16Bn_PWM6 输出状态。	R/W	0
5	EM5	TC 和 MR5 相等时, 该位根据 EMC5 位进行动作, 并驱动 CT16Bn_PWM5 输出状态。	R/W	0
4	EM4	TC 和 MR4 相等时, 该位根据 EMC4 位进行动作, 并驱动 CT16Bn_PWM4 输出状态。	R/W	0
3	EM3	TC 和 MR3 相等时, 该位根据 EMC3 位进行动作, 并驱动 CT16Bn_PWM3 输出状态。	R/W	0
2	EM2	TC 和 MR2 相等时, 该位根据 EMC2 位进行动作, 并驱动 CT16Bn_PWM2 输出状态。	R/W	0
1	EM1	TC 和 MR1 相等时, 该位根据 EMC1 位进行动作, 并驱动 CT16Bn_PWM1 输出状态。	R/W	0
0	EM0	TC 和 MR0 相等时, 该位根据 EMC0 位进行动作, 并驱动 CT16Bn_PWM0 输出状态。	R/W	0

### 6.7.15 CT16Bn外部匹配控制寄存器（CT16Bn\_EMC）（n=1）

地址偏移量：0x8C

外部匹配寄存器提供 CT16Bn\_PWM[22:0]的控制和状态。若匹配输出配置为 PWM 输出，外部匹配寄存器的功能由 PWM 规则决定。

Bit	Name	Description	Attribute	Reset
31:30	EMC15[1:0]	CT16Bn_PWM15 功能决定位。 00: 无功能; 01: CT16Bn_PWM15 引脚为低电平; 10: CT16Bn_PWM15 引脚为高电平; 11: 反转 CT16Bn_PWM15 引脚状态。	R/W	0
29:28	EMC14[1:0]	CT16Bn_PWM14 功能决定位。 00: 无功能; 01: CT16Bn_PWM14 引脚为低电平; 10: CT16Bn_PWM14 引脚为高电平; 11: 反转 CT16Bn_PWM14 引脚状态。	R/W	0
27:26	EMC13[1:0]	CT16Bn_PWM13 功能决定位。 00: 无功能; 01: CT16Bn_PWM13 引脚为低电平; 10: CT16Bn_PWM13 引脚为高电平; 11: 反转 CT16Bn_PWM13 引脚状态。	R/W	0
25:24	EMC12[1:0]	CT16Bn_PWM12 功能决定位。 00: 无功能; 01: CT16Bn_PWM12 引脚为低电平; 10: CT16Bn_PWM12 引脚为高电平; 11: 反转 CT16Bn_PWM12 引脚状态。	R/W	0
23:22	EMC11[1:0]	CT16Bn_PWM11 功能决定位。 00: 无功能; 01: CT16Bn_PWM11 引脚为低电平; 10: CT16Bn_PWM11 引脚为高电平; 11: 反转 CT16Bn_PWM11 引脚状态。	R/W	0
21:20	EMC10[1:0]	CT16Bn_PWM10 功能决定位。 00: 无功能; 01: CT16Bn_PWM10 引脚为低电平; 10: CT16Bn_PWM10 引脚为高电平; 11: 反转 CT16Bn_PWM10 引脚状态。	R/W	0
19:18	EMC9[1:0]	CT16Bn_PWM9 功能决定位。 00: 无功能; 01: CT16Bn_PWM9 引脚为低电平; 10: CT16Bn_PWM9 引脚为高电平; 11: 反转 CT16Bn_PWM9 引脚状态。	R/W	0
17:16	EMC8[1:0]	CT16Bn_PWM8 功能决定位。 00: 无功能; 01: CT16Bn_PWM8 引脚为低电平; 10: CT16Bn_PWM8 引脚为高电平; 11: 反转 CT16Bn_PWM8 引脚状态。	R/W	0
15:14	EMC7[1:0]	CT16Bn_PWM7 功能决定位。 00: 无功能; 01: CT16Bn_PWM7 引脚为低电平; 10: CT16Bn_PWM7 引脚为高电平; 11: 反转 CT16Bn_PWM7 引脚状态。	R/W	0
13:12	EMC6[1:0]	CT16Bn_PWM6 功能决定位。 00: 无功能; 01: CT16Bn_PWM6 引脚为低电平; 10: CT16Bn_PWM6 引脚为高电平; 11: 反转 CT16Bn_PWM6 引脚状态。	R/W	0
11:10	EMC5[1:0]	CT16Bn_PWM5 功能决定位。 00: 无功能; 01: CT16Bn_PWM5 引脚为低电平; 10: CT16Bn_PWM5 引脚为高电平;	R/W	0

		11: 反转 CT16Bn_PWM5 引脚状态。		
9:8	EMC4[1:0]	CT16Bn_PWM4 功能决定位。 00: 无功能; 01: CT16Bn_PWM4 引脚为低电平; 10: CT16Bn_PWM4 引脚为高电平; 11: 反转 CT16Bn_PWM4 引脚状态。	R/W	0
7:6	EMC3[1:0]	CT16Bn_PWM3 功能决定位。 00: 无功能; 01: CT16Bn_PWM3 引脚为低电平; 10: CT16Bn_PWM3 引脚为高电平; 11: 反转 CT16Bn_PWM3 引脚状态。	R/W	0
5:4	EMC2[1:0]	CT16Bn_PWM2 功能决定位。 00: 无功能; 01: CT16Bn_PWM2 引脚为低电平; 10: CT16Bn_PWM2 引脚为高电平; 11: 反转 CT16Bn_PWM2 引脚状态。	R/W	0
3:2	EMC1[1:0]	CT16Bn_PWM1 功能决定位。 00: 无功能; 01: CT16Bn_PWM1 引脚为低电平; 10: CT16Bn_PWM1 引脚为高电平; 11: 反转 CT16Bn_PWM1 引脚状态。	R/W	0
1:0	EMC0[1:0]	CT16Bn_PWM0 功能决定位。 00: 无功能; 01: CT16Bn_PWM0 引脚为低电平; 10: CT16Bn_PWM0 引脚为高电平; 11: 反转 CT16Bn_PWM0 引脚状态。	R/W	0



## 6.7.16 CT16Bn外部匹配控制寄存器 2 (CT16Bn\_EMC2) (n=1)

地址偏移量: 0x90

外部匹配寄存器提供 CT16Bn\_PWM[22:0]的控制和状态。若匹配输出配置为 PWM 输出, 外部匹配寄存器的功能由 PWM 规则决定。

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:12	EMC22[1:0]	CT16Bn_PWM22 功能决定位。 00: 无功能; 01: CT16Bn_PWM22 引脚为低电平; 10: CT16Bn_PWM22 引脚为高电平; 11: 反转 CT16Bn_PWM22 引脚状态。	R/W	0
11:10	EMC21[1:0]	CT16Bn_PWM21 功能决定位。 00: 无功能; 01: CT16Bn_PWM21 引脚为低电平; 10: CT16Bn_PWM21 引脚为高电平; 11: 反转 CT16Bn_PWM12 引脚状态。	R/W	0
9:8	Reserved		R	0
7:6	EMC19[1:0]	CT16Bn_PWM19 功能决定位。 00: 无功能; 01: CT16Bn_PWM19 引脚为低电平; 10: CT16Bn_PWM19 引脚为高电平; 11: 反转 CT16Bn_PWM19 引脚状态。	R/W	0
5:4	EMC18[1:0]	CT16Bn_PWM18 功能决定位。 00: 无功能; 01: CT16Bn_PWM18 引脚为低电平; 10: CT16Bn_PWM18 引脚为高电平; 11: 反转 CT16Bn_PWM18 引脚状态。	R/W	0
3:2	EMC17[1:0]	CT16Bn_PWM17 功能决定位。 00: 无功能; 01: CT16Bn_PWM17 引脚为低电平; 10: CT16Bn_PWM17 引脚为高电平; 11: 反转 CT16Bn_PWM17 引脚状态。	R/W	0
1:0	EMC16[1:0]	CT16Bn_PWM16 功能决定位。 00: 无功能; 01: CT16Bn_PWM16 引脚为低电平; 10: CT16Bn_PWM16 引脚为高电平; 11: 反转 CT16Bn_PWM16 引脚状态。	R/W	0

### 6.7.17 CT16Bn PWM控制寄存器 (CT16Bn\_PWMCTRL) (n=1)

地址偏移量: 0x94

PWM 控制寄存器用于配置匹配输出作为 PWM 输出。每个匹配输出都可独立的作为 PWM 输出或者匹配输出执行，其功能由 CT16Bn\_EM 寄存器控制。

对于每个定时器来说,最多控制 4 路 PWM 输出,由 CT16Bn\_PWMCTRL[3:0]选择输出。额外的匹配寄存器决定 PWM 周期长度。在其它的匹配寄存器发生匹配时, PWM 输出高电平。定时器由设置 PWM 周期长度的匹配寄存器复位。定时器复位为 0 时,将当前配置为 PWM 输出的高电平输出都清零。

Bit	Name	Description	Attribute	Reset
31:30	PWM15MODE[1:0]	PWM15 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR15, PWM15 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR15, PWM15 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
29:28	PWM14MODE[1:0]	PWM14 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR14, PWM14 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR14, PWM14 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
27:26	PWM13MODE[1:0]	PWM13 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR13, PWM13 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR13, PWM13 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
25:24	PWM12MODE[1:0]	PWM12 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR12, PWM12 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR12, PWM12 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
23:22	PWM11MODE[1:0]	PWM11 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR11, PWM11 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR11, PWM11 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
21:20	PWM10MODE[1:0]	PWM10 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR10, PWM10 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR10, PWM10 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
19:18	PWM9MODE[1:0]	PWM9 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR9, PWM9 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR9, PWM9 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
17:16	PWM8MODE[1:0]	PWM8 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR8, PWM8 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR8, PWM8 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
15:14	PWM7MODE[1:0]	PWM7 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR7, PWM7 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR7, PWM7 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
13:12	PWM6MODE[1:0]	PWM6 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR6, PWM6 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR6, PWM6 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0

11:10	PWM5MODE[1:0]	PWM5 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 $TC < MR5$ , PWM5 为 0; 01: PWM 模式 2, 向上计数期间, 若 $TC < MR5$ , PWM5 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
9:8	PWM4MODE[1:0]	PWM4 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 $TC < MR4$ , PWM4 为 0; 01: PWM 模式 2, 向上计数期间, 若 $TC < MR4$ , PWM4 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
7:6	PWM3MODE[1:0]	PWM3 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 $TC < MR3$ , PWM3 为 0; 01: PWM 模式 2, 向上计数期间, 若 $TC < MR3$ , PWM3 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
5:4	PWM2MODE[1:0]	PWM2 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 $TC < MR2$ , PWM2 为 0; 01: PWM 模式 2, 向上计数期间, 若 $TC < MR2$ , PWM2 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
3:2	PWM1MODE[1:0]	PWM1 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 $TC < MR1$ , PWM1 为 0; 01: PWM 模式 2, 向上计数期间, 若 $TC < MR1$ , PWM1 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
1:0	PWM0MODE[1:0]	PWM0 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 $TC < MR0$ , PWM0 为 0; 01: PWM 模式 2, 向上计数期间, 若 $TC < MR0$ , PWM0 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0

## 6.7.18 CT16Bn PWM控制寄存器 2 (CT16Bn\_PWMCTRL2) (n=1)

地址偏移量: 0x98

PWM 控制寄存器用于配置匹配输出作为 PWM 输出。每个匹配输出都可独立的作为 PWM 输出或者匹配输出执行，其功能由 CT16Bn\_EM 寄存器控制。

对于每个定时器来说,最多控制 4 路 PWM 输出,由 CT16Bn\_PWMCTRL[3:0]选择输出。额外的匹配寄存器决定 PWM 周期长度。在其它的匹配寄存器发生匹配时, PWM 输出高电平。定时器由设置 PWM 周期长度的匹配寄存器复位。定时器复位为 0 时,将当前配置为 PWM 输出的高电平输出都清零。

Bit	Name	Description	Attribute	Reset
31:14	Reserved		R	0
13:12	PWM22MODE[1:0]	PWM22 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR22, PWM22 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR22, PWM22 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
11:10	PWM21MODE[1:0]	PWM21 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR21, PWM21 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR21, PWM21 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
9:8	Reserved		R	0
7:6	PWM19MODE[1:0]	PWM19 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR19, PWM19 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR19, PWM19 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
5:4	PWM18MODE[1:0]	PWM18 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR18, PWM18 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR18, PWM18 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
3:2	PWM17MODE[1:0]	PWM17 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR17, PWM17 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR17, PWM17 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0
1:0	PWM16MODE[1:0]	PWM16 输出模式控制位。 00: PWM 模式 1, 向上计数期间, 若 TC<MR16, PWM16 为 0; 01: PWM 模式 2, 向上计数期间, 若 TC<MR16, PWM16 为 1 10: 强制为 0; 11: 强制为 1。	R/W	0

### 6.7.19 CT16Bn PWM使能寄存器 (CT16Bn\_PWMENB) (n=1)

地址偏移量: 0x9C

PWM 控制寄存器用于配置匹配输出作为 PWM 输出。每个匹配输出都可独立的作为 PWM 输出或者匹配输出执行，其功能由 CT16Bn\_EM 寄存器控制。

对于每个定时器来说,最多控制 4 路 PWM 输出,由 CT16Bn\_PWMCTRL[3:0]选择输出。额外的匹配寄存器决定 PWM 周期长度。在其它的匹配寄存器发生匹配时, PWM 输出高电平。定时器由设置 PWM 周期长度的匹配寄存器复位。定时器复位为 0 时,将当前配置为 PWM 输出的高电平输出都清零。

Bit	Name	Description	Attribute	Reset
31:23	Reserved		R	0
22	PWM22EN	PWM22 使能位。 0: CT16Bn_PWM22 由 EM22 控制; 1: 使能 CT16Bn_PWM22 的 PWM 模式。	R/W	0
21	PWM21EN	PWM21 使能位。 0: CT16Bn_PWM21 由 EM21 控制; 1: 使能 CT16Bn_PWM21 的 PWM 模式。	R/W	0
20	Reserved		R	0
19	PWM19EN	PWM19 使能位。 0: CT16Bn_PWM19 由 EM19 控制; 1: 使能 CT16Bn_PWM19 的 PWM 模式。	R/W	0
18	PWM18EN	PWM18 使能位。 0: CT16Bn_PWM18 由 EM18 控制; 1: 使能 CT16Bn_PWM18 的 PWM 模式。	R/W	0
17	PWM17EN	PWM17 使能位。 0: CT16Bn_PWM17 由 EM17 控制; 1: 使能 CT16Bn_PWM17 的 PWM 模式。	R/W	0
16	PWM16EN	PWM16 使能位。 0: CT16Bn_PWM16 由 EM16 控制; 1: 使能 CT16Bn_PWM16 的 PWM 模式。	R/W	0
15	PWM15EN	PWM15 使能位。 0: CT16Bn_PWM15 由 EM15 控制; 1: 使能 CT16Bn_PWM15 的 PWM 模式。	R/W	0
14	PWM14EN	PWM14 使能位。 0: CT16Bn_PWM14 由 EM14 控制; 1: 使能 CT16Bn_PWM14 的 PWM 模式。	R/W	0
13	PWM13EN	PWM13 使能位。 0: CT16Bn_PWM13 由 EM13 控制; 1: 使能 CT16Bn_PWM13 的 PWM 模式。	R/W	0
12	PWM12EN	PWM12 使能位。 0: CT16Bn_PWM12 由 EM12 控制; 1: 使能 CT16Bn_PWM12 的 PWM 模式。	R/W	0
11	PWM11EN	PWM11 使能位。 0: CT16Bn_PWM11 由 EM11 控制; 1: 使能 CT16Bn_PWM11 的 PWM 模式。	R/W	0
10	PWM10EN	PWM10 使能位。 0: CT16Bn_PWM10 由 EM10 控制; 1: 使能 CT16Bn_PWM10 的 PWM 模式。	R/W	0
9	PWM9EN	PWM9 使能位。 0: CT16Bn_PWM9 由 EM9 控制; 1: 使能 CT16Bn_PWM9 的 PWM 模式。	R/W	0
8	PWM8EN	PWM8 使能位。 0: CT16Bn_PWM8 由 EM8 控制; 1: 使能 CT16Bn_PWM8 的 PWM 模式。	R/W	0
7	PWM7EN	PWM7 使能位。 0: CT16Bn_PWM7 由 EM7 控制; 1: 使能 CT16Bn_PWM7 的 PWM 模式。	R/W	0
6	PWM6EN	PWM6 使能位。 0: CT16Bn_PWM6 由 EM6 控制; 1: 使能 CT16Bn_PWM6 的 PWM 模式。	R/W	0

5	PWM5EN	PWM5 使能位。 0: CT16Bn_PWM5 由 EM5 控制; 1: 使能 CT16Bn_PWM5 的 PWM 模式。	R/W	0
4	PWM4EN	PWM4 使能位。 0: CT16Bn_PWM4 由 EM4 控制; 1: 使能 CT16Bn_PWM4 的 PWM 模式。	R/W	0
3	PWM3EN	PWM3 使能位。 0: CT16Bn_PWM3 由 EM3 控制; 1: 使能 CT16Bn_PWM3 的 PWM 模式。	R/W	0
2	PWM2EN	PWM2 使能位。 0: CT16Bn_PWM2 由 EM2 控制; 1: 使能 CT16Bn_PWM2 的 PWM 模式。	R/W	0
1	PWM1EN	PWM1 使能位。 0: CT16Bn_PWM1 由 EM1 控制; 1: 使能 CT16Bn_PWM1 的 PWM 模式。	R/W	0
0	PWM0EN	PWM0 使能位。 0: CT16Bn_PWM0 由 EM0 控制; 1: 使能 CT16Bn_PWM0 的 PWM 模式。	R/W	0

## 6.7.20 PWM IO使能寄存器 (CT16Bn\_PWMIOENB) (n=1)

地址偏移量: 0Xa0

PWM 控制寄存器用于配置匹配输出作为 PWM 输出。每个匹配输出都可独立的作为 PWM 输出或者匹配输出执行，其功能由 CT16Bn\_EM 寄存器控制。

对于每个定时器来说，最多控制 4 路 PWM 输出，由 CT16Bn\_PWMCTRL[3:0]选择输出。额外的匹配寄存器决定 PWM 周期长度。在其它的匹配寄存器发生匹配时，PWM 输出高电平。定时器由设置 PWM 周期长度的匹配寄存器复位。定时器复位为 0 时，将当前配置为 PWM 输出的高电平输出都清零。

Bit	Name	Description	Attribute	Reset
31:23	Reserved		R	0
22	PWM22IOEN	CT16Bn_PWM22/GPIO 选择位。 0: CT16Bn_PWM22 引脚作为 GPIO 引脚; 1: CT16Bn_PWM22 引脚作为匹配输出, 其输出信号取决于 PWM22EN 位。	R/W	0
21	PWM21IOEN	CT16Bn_PWM21/GPIO 选择位。 0: CT16Bn_PWM21 引脚作为 GPIO 引脚; 1: CT16Bn_PWM21 引脚作为匹配输出, 其输出信号取决于 PWM21EN 位。	R/W	0
20	Reserved		R	0
19	PWM19IOEN	CT16Bn_PWM19/GPIO 选择位。 0: CT16Bn_PWM19 引脚作为 GPIO 引脚; 1: CT16Bn_PWM19 引脚作为匹配输出, 其输出信号取决于 PWM19EN 位。	R/W	0
18	PWM18IOEN	CT16Bn_PWM18/GPIO 选择位。 0: CT16Bn_PWM18 引脚作为 GPIO 引脚; 1: CT16Bn_PWM18 引脚作为匹配输出, 其输出信号取决于 PWM18EN 位。	R/W	0
17	PWM17IOEN	CT16Bn_PWM17/GPIO 选择位。 0: CT16Bn_PWM17 引脚作为 GPIO 引脚; 1: CT16Bn_PWM17 引脚作为匹配输出, 其输出信号取决于 PWM17EN 位。	R/W	0
16	PWM16IOEN	CT16Bn_PWM16/GPIO 选择位。 0: CT16Bn_PWM16 引脚作为 GPIO 引脚; 1: CT16Bn_PWM16 引脚作为匹配输出, 其输出信号取决于 PWM16EN 位。	R/W	0
15	PWM15IOEN	CT16Bn_PWM15/GPIO 选择位。 0: CT16Bn_PWM15 引脚作为 GPIO 引脚; 1: CT16Bn_PWM15 引脚作为匹配输出, 其输出信号取决于 PWM15EN 位。	R/W	0
14	PWM14IOEN	CT16Bn_PWM14/GPIO 选择位。 0: CT16Bn_PWM14 引脚作为 GPIO 引脚; 1: CT16Bn_PWM14 引脚作为匹配输出, 其输出信号取决于 PWM14EN 位。	R/W	0
13	PWM13IOEN	CT16Bn_PWM13/GPIO 选择位。 0: CT16Bn_PWM13 引脚作为 GPIO 引脚; 1: CT16Bn_PWM13 引脚作为匹配输出, 其输出信号取决于 PWM13EN 位。	R/W	0
12	PWM12IOEN	CT16Bn_PWM12/GPIO 选择位。 0: CT16Bn_PWM12 引脚作为 GPIO 引脚; 1: CT16Bn_PWM12 引脚作为匹配输出, 其输出信号取决于 PWM12EN 位。	R/W	0
11	PWM11IOEN	CT16Bn_PWM11/GPIO 选择位。 0: CT16Bn_PWM11 引脚作为 GPIO 引脚; 1: CT16Bn_PWM11 引脚作为匹配输出, 其输出信号取决于 PWM11EN 位。	R/W	0
10	PWM10IOEN	CT16Bn_PWM10/GPIO 选择位。 0: CT16Bn_PWM10 引脚作为 GPIO 引脚; 1: CT16Bn_PWM10 引脚作为匹配输出, 其输出信号取决于 PWM10EN 位。	R/W	0
9	PWM9IOEN	CT16Bn_PWM9/GPIO 选择位。 0: CT16Bn_PWM9 引脚作为 GPIO 引脚; 1: CT16Bn_PWM9 引脚作为匹配输出, 其输出信号取决于 PWM9EN 位。	R/W	0
8	PWM8IOEN	CT16Bn_PWM8/GPIO 选择位。 0: CT16Bn_PWM8 引脚作为 GPIO 引脚; 1: CT16Bn_PWM8 引脚作为匹配输出, 其输出信号取决于 PWM8EN 位。	R/W	0
7	PWM7IOEN	CT16Bn_PWM7/GPIO 选择位。 0: CT16Bn_PWM7 引脚作为 GPIO 引脚; 1: CT16Bn_PWM7 引脚作为匹配输出, 其输出信号取决于 PWM7EN 位。	R/W	0
6	PWM6IOEN	CT16Bn_PWM6/GPIO 选择位。 0: CT16Bn_PWM6 引脚作为 GPIO 引脚; 1: CT16Bn_PWM6 引脚作为匹配输出, 其输出信号取决于 PWM6EN 位。	R/W	0



5	PWM5IOEN	CT16Bn_PWM5/GPIO 选择位。 0: CT16Bn_PWM5 引脚作为 GPIO 引脚； 1: CT16Bn_PWM5 引脚作为匹配输出，其输出信号取决于 PWM5EN 位。	R/W	0
4	PWM4IOEN	CT16Bn_PWM4/GPIO 选择位。 0: CT16Bn_PWM4 引脚作为 GPIO 引脚； 1: CT16Bn_PWM4 引脚作为匹配输出，其输出信号取决于 PWM4EN 位。	R/W	0
3	PWM3IOEN	CT16Bn_PWM3/GPIO 选择位。 0: CT16Bn_PWM3 引脚作为 GPIO 引脚； 1: CT16Bn_PWM3 引脚作为匹配输出，其输出信号取决于 PWM3EN 位。	R/W	0
2	PWM2IOEN	CT16Bn_PWM2/GPIO 选择位。 0: CT16Bn_PWM2 引脚作为 GPIO 引脚； 1: CT16Bn_PWM2 引脚作为匹配输出，其输出信号取决于 PWM2EN 位。	R/W	0
1	PWM1IOEN	CT16Bn_PWM1/GPIO 选择位。 0: CT16Bn_PWM1 引脚作为 GPIO 引脚； 1: CT16Bn_PWM1 引脚作为匹配输出，其输出信号取决于 PWM1EN 位。	R/W	0
0	PWM0IOEN	CT16Bn_PWM0/GPIO 选择位。 0: CT16Bn_PWM0 引脚作为 GPIO 引脚； 1: CT16Bn_PWM0 引脚作为匹配输出，其输出信号取决于 PWM0EN 位。	R/W	0



## 6.7.21 CT16Bn定时器中断源状态寄存器（CT16Bn\_RIS）（n=0,1）

地址偏移量：0xA4

该寄存器显示定时器/PWM 中断的原始状态，若 CT16Bn\_IE 寄存器的相应位置 1，定时器/PWM 中断发送到中断控制器。

Bit	Name	Description	Attribute	Reset
31:25	Reserved		R	0
24	CAP0IF	捕获通道 0 的中断标志位。 0: CAP0 无中断; 1: CAP0 发生中断请求。	R	0
23	MR23IF	匹配通道 23 的中断标志位。 0: 匹配通道 23 无中断; 1: 匹配中断 23 发生中断请求。	R	0
22	MR22IF	匹配通道 22 的中断标志位。 0: 匹配通道 22 无中断; 1: 匹配中断 22 发生中断请求。	R	0
21	MR21IF	匹配通道 21 的中断标志位。 0: 匹配通道 21 无中断; 1: 匹配中断 21 发生中断请求。	R	0
20	Reserved		R	0
19	MR19IF	匹配通道 19 的中断标志位。 0: 匹配通道 19 无中断; 1: 匹配中断 19 发生中断请求。	R	0
18	MR18IF	匹配通道 18 的中断标志位。 0: 匹配通道 18 无中断; 1: 匹配中断 18 发生中断请求。	R	0
17	MR17IF	匹配通道 17 的中断标志位。 0: 匹配通道 17 无中断; 1: 匹配中断 17 发生中断请求。	R	0
16	MR16IF	匹配通道 16 的中断标志位。 0: 匹配通道 16 无中断; 1: 匹配中断 16 发生中断请求。	R	0
15	MR15IF	匹配通道 15 的中断标志位。 0: 匹配通道 15 无中断; 1: 匹配中断 15 发生中断请求。	R	0
14	MR14IF	匹配通道 14 的中断标志位。 0: 匹配通道 14 无中断; 1: 匹配中断 14 发生中断请求。	R	0
13	MR13IF	匹配通道 13 的中断标志位。 0: 匹配通道 13 无中断; 1: 匹配中断 13 发生中断请求。	R	0
12	MR12IF	匹配通道 12 的中断标志位。 0: 匹配通道 12 无中断; 1: 匹配中断 12 发生中断请求。	R	0
11	MR11IF	匹配通道 11 的中断标志位。 0: 匹配通道 11 无中断; 1: 匹配中断 11 发生中断请求。	R	0
10	MR10IF	匹配通道 10 的中断标志位。 0: 匹配通道 10 无中断; 1: 匹配中断 10 发生中断请求。	R	0
9	MR9IF	匹配通道 9 的中断标志位。 0: 匹配通道 9 无中断; 1: 匹配中断 9 发生中断请求。	R	0
8	MR8IF	匹配通道 8 的中断标志位。 0: 匹配通道 8 无中断; 1: 匹配中断 8 发生中断请求。	R	0
7	MR7IF	匹配通道 7 的中断标志位。 0: 匹配通道 7 无中断; 1: 匹配中断 7 发生中断请求。	R	0
6	MR6IF	匹配通道 6 的中断标志位。	R	0

		0: 匹配通道 6 无中断; 1: 匹配中断 6 发生中断请求。		
5	MR5IF	匹配通道 5 的中断标志位。 0: 匹配通道 5 无中断; 1: 匹配中断 5 发生中断请求。	R	0
4	MR4IF	匹配通道 4 的中断标志位。 0: 匹配通道 4 无中断; 1: 匹配中断 4 发生中断请求。	R	0
3	MR3IF	匹配通道 3 的中断标志位。 0: 匹配通道 3 无中断; 1: 匹配中断 3 发生中断请求。	R	0
2	MR2IF	匹配通道 2 的中断标志位。 0: 匹配通道 2 无中断; 1: 匹配中断 2 发生中断请求。	R	0
1	MR1IF	匹配通道 1 的中断标志位。 0: 匹配通道 1 无中断; 1: 匹配中断 1 发生中断请求。	R	0
0	MR0IF	匹配通道 0 的中断标志位。 0: 匹配通道 0 无中断; 1: 匹配中断 0 发生中断请求。	R	0

## 6.7.22 CT16Bn定时器中断清零寄存器 (CT16Bn\_IC) (n=0,1)

地址偏移量: 0xA8

Bit	Name	Description	Attribute	Reset
31:25	Reserved		R	0
24	CAP0IC	0: 无影响; 1: CAP0IF 位清零。	W	0
23	MR23IC	0: 无影响; 1: MR23IF 位清零。	W	0
22	MR22IC	0: 无影响; 1: MR22IF 位清零。	W	0
21	MR21IC	0: 无影响; 1: MR21IF 位清零。	W	0
20	Reserved		R	0
19	MR19IC	0: 无影响; 1: MR19IF 位清零。	W	0
18	MR18IC	0: 无影响; 1: MR18IF 位清零。	W	0
17	MR17IC	0: 无影响; 1: MR17IF 位清零。	W	0
16	MR16IC	0: 无影响; 1: MR16IF 位清零。	W	0
15	MR15IC	0: 无影响; 1: MR15IF 位清零。	W	0
14	MR14IC	0: 无影响; 1: MR14IF 位清零。	W	0
13	MR13IC	0: 无影响; 1: MR13IF 位清零。	W	0
12	MR12IC	0: 无影响; 1: MR12IF 位清零。	W	0
11	MR11IC	0: 无影响; 1: MR11IF 位清零。	W	0
10	MR10IC	0: 无影响; 1: MR10IF 位清零。	W	0
9	MR9IC	0: 无影响; 1: MR9IF 位清零。	W	0
8	MR8IC	0: 无影响; 1: MR8IF 位清零。	W	0
7	MR7IC	0: 无影响; 1: MR7IF 位清零。	W	0
6	MR6IC	0: 无影响; 1: MR6IF 位清零。	W	0
5	MR5IC	0: 无影响; 1: MR5IF 位清零。	W	0
4	MR4IC	0: 无影响; 1: MR4IF 位清零。	W	0
3	MR3IC	0: 无影响; 1: MR3IF 位清零。	W	0
2	MR2IC	0: 无影响; 1: MR2IF 位清零。	W	0
1	MR1IC	0: 无影响; 1: MR1IF 位清零。	W	0
0	MR0IC	0: 无影响; 1: MR0IF 位清零。	W	0

# 7 看门狗定时器 (WDT)

## 7.1 概述

若系统进入了错误的工作状态，看门狗可以在合理的时间范围内使单片机复位。使能看门狗时，若用户程序清除看门狗定时器失败，则在预订的时间范围内，看门狗会使系统复位或者产生一个中断。

看门狗由 128 个预分频的分频器和 8 位计数器组成，时钟经由预分频提供给定时器。定时器递减计数，计数器递减到 0x01 时为最小值。因此，看门狗的最小间隔为  $T_{WDT\_PCLK} \times 128 \times 1$ ，最大间隔为  $T_{WDT\_PCLK} \times 128 \times 256$

看门狗按照下列方法进行操作：

- 1、通过 WDTCLKSEL 寄存器为看门狗选择时钟源；
- 2、通过 APB 时钟预分频寄存器 1 (SYS1\_APBCLKP1) 的 WDTPRE 位为看门狗设置预分频值；
- 3、在 WDT\_TC 寄存器中设置看门狗定时器的恒定重装值；
- 4、使能看门狗，通过 WDT\_CFG 寄存器设置看门狗定时器的操作模式；
- 5、通过写入 0x55AA 到 WDT\_FEED 寄存器清除看门狗，以防止看门狗计数器溢出引起复位或产生中断

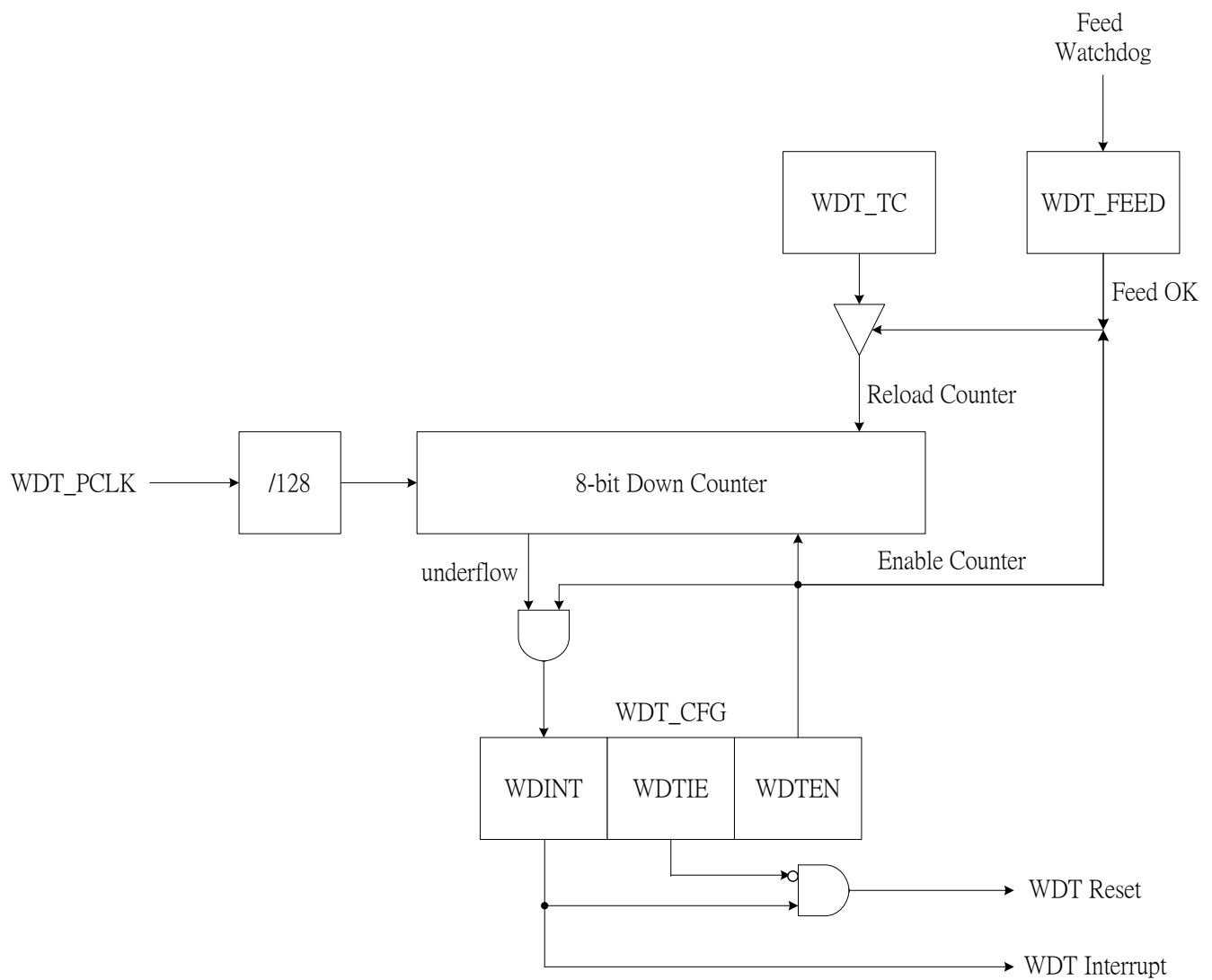
设置 WDT\_CFG 寄存器的 WDTEN 位时，看门狗开始运行，时间恒定值装入看门狗计数器，计数器开始递减计数。看门狗在复位模式下工作时，计数器溢出后，CPU 复位，随着外部复位，系统从向量表中将数据导入堆栈指针和程序计数器中。不管何时写入 0x55AA 到 WDT\_FEED 寄存器中，WDT\_TC 的值装入看门狗计数器，并阻止看门狗复位或者中断。

看门狗定时器的时钟有两种：HCLK 和 WDT\_PCLK。HCLK 用于 AHB 访问看门狗寄存器，并由系统时钟衍生；WDT\_PCLK 用于看门狗定时器计数。下列几种时钟都可用作 WDT\_PCLK 的时钟源：IHRC，ILRC 和 HCLK。

在 AHB 时钟使能寄存器 (SYS1\_AHBCLKEN) 中禁止看门狗寄存器模块以省电。

当看门狗正在运行时，任何时候都可能发生看门狗中断或复位

## 7.2 框图



## 7.3 WDT寄存器

基地址：0x4001 0000

### 7.3.1 看门狗配置寄存器（WDT\_CFG）

地址偏移量：0x00

WDT\_CFG 寄存器通过 WDTEN 和 WDTIE 位控制看门狗的操作。该寄存器显示看门狗定时器中断的 Raw 状态。若 WDTINT 和 WDTIE 位都置 1，则 WDT 中断发送到中断控制器中。

Bit	Name	Description	Attribute	Reset
31:16	WDKEY	看门狗寄存器按键。 读取为 0。执行写动作时必须写入 0x5AFA 到寄存器的 WDKEY 位，否则执行的写动作会被忽略。	W	0
15:3	Reserved		R	0
2	WDTINT	看门狗中断标志位。 <Read> 0: 看门狗没有引起中断； 1: 看门狗超时，引起中断（当且仅当 WDTIE=1）。 <Write> 0: 将该位清零，清零之前 SW 会重新设置看门狗。	R/W	0
1	WDTIE	看门狗中断使能位。 0: 看门狗超时，引起芯片复位（看门狗复位模式）；看门狗计数器下溢引起单片机复位，并将 WDTINT 位清零； 1: 看门狗超时引起中断（看门狗中断模式）。	R/W	0
0	WDTEN	看门狗使能位。 0: 禁止； 1: 使能，使能看门狗时，WDT_TC 的值装入看门狗计数器。	R/W	0

### 7.3.2 看门狗定时器常量寄存器（WDT\_TC）

地址偏移量：0x08

WDT\_TC 寄存器决定超时的时间，每次清看门狗，都会重装 WDT\_TC 的数值到看门狗定时器中。它是一个 8 位计数器，定时器的超时时间为  $T_{WDT\_PCLK} \times 128 \times 1 \sim T_{WDT\_PCLK} \times 128 \times 256$ 。

看门狗的溢出时间 =  $(31.25\mu s \times 1) \times 128 \times 1 \sim (31.25\mu s \times 32) \times 128 \times 256 = 4ms \sim 32768ms$

Bit	Name	Description	Attribute	Reset
31:16	WDKEY	看门狗寄存器按键。 读取为 0。执行写动作时必须写入 0x5AFA 到寄存器的 WDKEY 位，否则执行的写动作会被忽略。	W	0
15:8	Reserved		R	0
7:0	TC[7:0]	看门狗定时器恒定重装值 = TC[7:0]+1 0000 0000: 定时器恒定值 = 1; 0000 0001: 定时器恒定值 = 2; ..... ..... 1111 1110: 定时器恒定值 = 255; 1111 1111: 定时器恒定值 = 256。	R/W	0xFF

看门狗定时器的时钟源为 ILRC。

### 7.3.3 看门狗Feed寄存器（WDT\_FEED）

地址偏移量：0x0C

Bit	Name	Description	Attribute	Reset
31:16	WDKEY	看门狗寄存器按键。 读取为 0。执行写动作时必须写入 0x5AFA 到寄存器的 WDKEY 位，否则执行的写动作会被忽略。	W	0
15:0	FV[15:0]	Feed 值（读取为 0） 0x55AA: 清看门狗，WDT_TC 的值重装到看门狗计数器。	W	0

# 8 SPI

## 8.1 概述

SPI 控制器可完成总线上多主机和多从机间的相互通讯。在数据传送过程中，只有单个主机和单个从机可以在总线上相互通讯。数据传送为全双工的 4~16 位的数据帧从主机流向从机，或者从从机流向主机。实际上，在这些数据流中通常只有一个携带着有意义的数据。

## 8.2 特性

- 兼容摩托罗拉 SPI。
- 同步串行通讯。
- 支持主机或从机操作。
- 8 帧 FIFO，用于发送或接收。
- 4~16 位帧。
- SPI 的最大速度为 24Mbps（主机）或 6Mbps（从机）。
- 数据的发送格式为 MSB 或者 LSB，由寄存器控制。
- 数据采样的起始相位位置可以选择第一个相位或者第二个相位，由寄存器控制。

## 8.3 引脚说明

Pin Name	Type	Description	GPIO Configuration
SCKn	O	SPI 串行时钟（主机）	
	I	SPI 串行时钟（从机）	取决于 GPIO <sub>n</sub> _CFG
SELn	O	SPI 从机选择/SSI 帧 Sync（主机）	
	I	SPI 从机选择（从机）	取决于 GPIO <sub>n</sub> _CFG
MISO <sub>n</sub>	I	主机输入，从机输出（主机）	取决于 GPIO <sub>n</sub> _CFG
	O	主机输入，从机输出（从机）	
MOSI <sub>n</sub>	O	主机输出，从机输入（主机）	
	I	主机输出，从机输入（从机）	取决于 GPIO <sub>n</sub> _CFG

## 8.4 接口说明

### 8.4.1 SPI

SPI 接口为 4 线接口，SSEL 信号作为一个从机信号进行选择。SPI 格式的主要特性是：空闲状态和 SCK 信号的相位都可通过 SSPn\_CTRL1 寄存器的 CPOL 和 CPHA 位编程控制。

CPOL 时钟极性控制位为低时，在 SCK 引脚生成稳定的低电平状态。CPOL 时钟极性控制位为高时，则在 CLK 引脚产生稳定的高电平状态，此时并没有发送数据。

CPHA 时钟相位位控制数据在哪个时钟相位上被采样。若 CPHA=1，SCK 的第一个边沿进行数据转换，SCK 的第二个边沿进行数据的接收和发送；若 CPHA=0，第一个位已经确定，在 SCK 的第一个边沿进行数据的接收和发送。

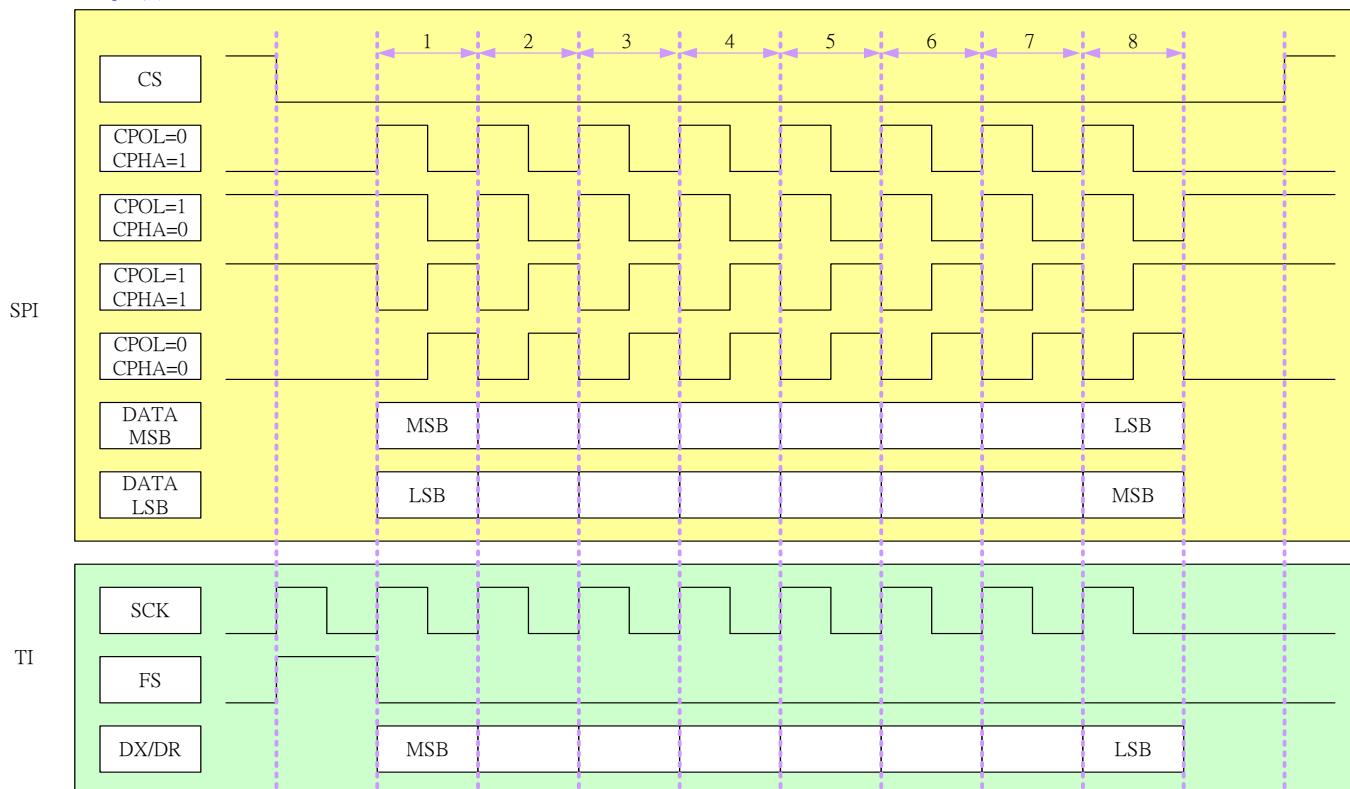
SIO 数据发送时序图如下：

MSLB	CPOL	CPHA	SCK Idle Status	Diagrams
0	0	1	Low	
0	1	1	High	
0	0	0	Low	
0	1	0	High	
1	0	1	Low	
1	1	1	High	
1	0	0	Low	
1	1	0	High	

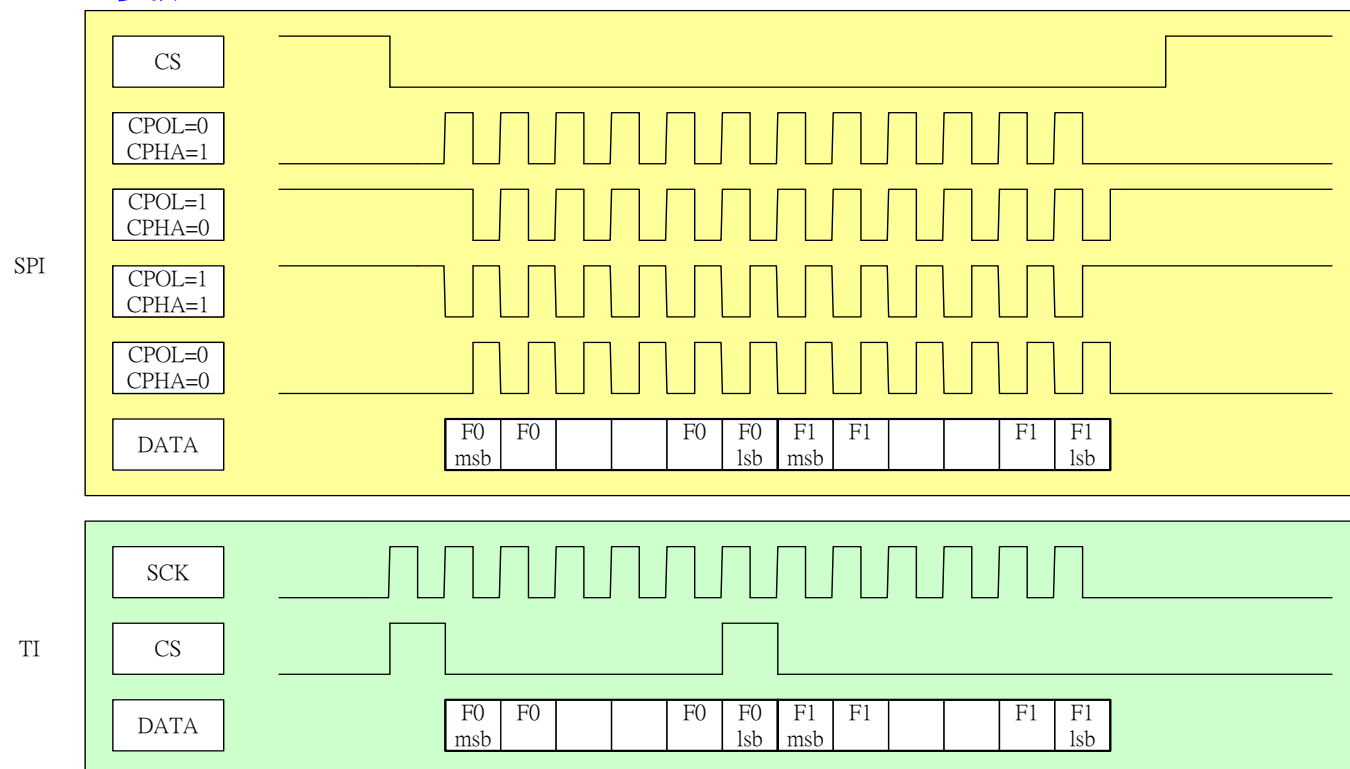


## 8.4.2 通讯流程

### 8.4.2.1 单帧



### 8.4.2.2 多帧



## 8.5 AUTO-SEL

Auto-SEL 功能的默认状态为禁止状态（SELDIS = 1），HW 不能控制 SELn 引脚，SELn 引脚为 GPIO 引脚。若使能 Auto-SEL 功能（SELDIS=0），SPI HW 控制 SELn 的状态，由 PFPA\_SSP 寄存器指定 SELn 引脚的功能。

## 8.6 SPI寄存器

基地址：0x4001 C000 (SSP0)

### 8.6.1 SPI n控制寄存器 0 (SPIn\_CTRL0) (n=0)

地址偏移量：0x00

\* 注：

1. 当 SSPEN=1 时，如果改变 SSP 的任何配置，都必须使用 FRESET[1:0]复位 SSP FSM。
2. 当 SSPEN=1 时，HW 直接切换 FORMAT 位决定的 I/O 配置。

Bit	Name	Description	Attribute	Reset
31:19	Reserved		R	0
18	SELDIS	Auto-SEL 功能禁止位，仅针对 SPI 模式。 0：使能 Auto-SEL 流程控制； 1：禁止 Auto-SEL 流程控制。	R/W	1
17:15	RXFIFOTH[2:0]	RX FIFO 阈电平。 000：RX FIFO 阈电平为 0； 001：RX FIFO 阈电平为 1； ... 111：RX FIFO 阈电平为 7。	R/W	000
14:12	TX FIFOTH[2:0]	TX FIFO 阈电平。 000：TX FIFO 阈电平为 0； 001：TX FIFO 阈电平为 1； ... 111：TX FIFO 阈电平为 7。	R/W	000
11:8	DL[3:0]	数据长度= DL[3:0] + 1。 0000~0001：保留； 0010：数据长度为 3； ... ... 1110：数据长度为 15； 1111：数据长度为 16。	R/W	1111b
7:6	FRESET[1:0]	SSP FSM 和 FIFO 复位位。 00：没有影响； 01：保留； 10：保留； 11：复位状态机和 FIFO（BUF_BUSY=0，移入 BUF 的数据清零，TX_EMPTY=1，TX_FULL=0，RX_EMPTY=1，RX_FULL=0，和 FIFO 的数据清零），该位由 HW 自动清零。	W	0
5	Reserved		R	0
4	FORMAT	接口格式。 0：SPI；1：保留。	R/W	0
3	MS	主机/从机选择位。 0：主机；1：从机。	R/W	0
2	SDODIS	从机数据输出禁止位（仅用于从机模式）。 0：使能从机数据输出； 1：禁止从机数据输出（MISO=0）。	R/W	0
1	LOOPBACK	回路模式使能位。 0：禁止；1：使能回路模式。	R/W	0
0	SPIEN	SPI 使能位。 0：禁止；1：使能，到达 FORMAT 位后 HW 直接切换 I/O 配置。	R/W	0

### 8.6.2 SPI n控制寄存器 1 (SPIn\_CTRL1) (n=0)

地址偏移量: 0x04

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	CPHA	边沿采样的时钟相位。 0: CPOL=0 时在时钟下降沿改变数据, 上升沿锁存, CPOL=1 时, 在时钟上升沿改变数据, 下降沿锁存; 1: CPOL=0 时在时钟上升沿改变数据, 下降沿锁存, CPOL=1 时, 在时钟下降沿改变数据, 上升沿锁存。	R/W	0
1	CPOL	时钟极性选择位。 0: SCK 空闲为低电平; 1: SCK 空闲为高电平。	R/W	0
0	MLSB	MSB/LSB 选择位。 0: 先发送高位; 1: 先发送低位。	R/W	0

### 8.6.3 SPI n时钟分频寄存器 (SPIn\_CLKDIV) (n=0)

地址偏移量: 0x08

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DIV[7:0]	SSPn 时钟分频。 0: SCK = SSPn_PCLK / 2; 1: SCK = SSPn_PCLK / 4; 2: SCK = SSPn_PCLK / 6; X: SCK = SSPn_PCLK / (2X+2)。	R/W	0

### 8.6.4 SPI n状态寄存器 (SPIn\_STAT) (n=0)

地址偏移量: 0x0C

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6	RXFIFOTHF	RX FIFO 临界值标志位。 0: Data in RX FIFO ≤ RXFIFOTH; 1: Data in RX FIFO > RXFIFOTH。	R	0
5	TXFIFOTHF	TX FIFO 临界值标志位。 0: Data in TX FIFO > TXFIFOTH; 1: Data in TX FIFO ≤ TXFIFOTH。	R	1
4	BUSY	繁忙标志位。 0: SSP 控制器空闲; 1: SSP 控制器正在工作中。	R	0
3	RX_FULL	RX FIFO 全满标志位。 0: RX FIFO 非全满; 1: RX FIFO 全满。	R	0
2	RX_EMPTY	RX FIFO 全空标志位。 0: RX FIFO 非全空; 1: RX FIFO 全空。	R	1
1	TX_FULL	TX FIFO 全满标志位。 0: TX FIFO 非全满; 1: TX FIFO 全满。	R	0
0	TX_EMPTY	TX FIFO 全空标志位。 0: TX FIFO 非全空, 主机模式下, 发送机自动开始发送; 1: TX FIFO 全空。	R	1

### 8.6.5 SPI n中断使能寄存器 (SPI<sub>n</sub>\_IE) (n=0)

地址偏移量: 0x10

该寄存器用于控制 SPI 控制器中 4 个中断的使能状态。

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	TXFIFOTHIE	TX FIFO 临界值中断使能位。 0: 禁止; 1: 使能。	R/W	0
2	RXFIFOTHIE	RX FIFO 临界值中断使能位。 0: 禁止; 1: 使能。	R/W	0
1	RXTOIE	RX 超时中断使能位。 0: 禁止; 1: 使能。	R/W	0
0	RXOVFIE	RX 溢出中断使能位。 0: 禁止; 1: 使能。	R/W	0

### 8.6.6 SPI n中断源状态寄存器 (SPI<sub>n</sub>\_RIS) (n=0)

地址偏移量: 0x14

该寄存器包含每个中断的状态, 不管是否在 SPI<sub>n</sub>\_IE 寄存器中使能中断。该寄存器显示 SPI 中断源状态。若 SPI<sub>n</sub>\_IE 寄存器的相应位置 1 时, 发送一个 SPI 中断到中断控制器中。

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	TXFIFOTHIF	TX FIFO 临界值中断标志位。 0: 没有 TX FIFO 临界值中断; 1: 触发 TX FIFO 临界值中断。	R	0
2	RXFIFOTHIF	RX FIFO 临界值中断标志位。 0: 没有 RX FIFO 临界值中断; 1: 触发 RX FIFO 临界值中断。	R	0
1	RXTOIF	RX 超时中断标志位。当 RX FIFO 不空时, 并且一定时间 (32*SSP <sub>n</sub> _PCLK) 没有被读出, 则发生 RXTO。这个溢出时间对主机模式和从机模式是一样。 0: 没有发生 RXTO; 1: 发生 RXTO。	R	0
0	RXOVFIF	RX 溢出中断标志位。当 RX FIFO 全满时, 并且另外一帧被完全接收, 则发生 RXOVF。ARM 规格描述当此种情况发生时, 新帧数据覆盖前帧数据。 0: 没有发生 RXOVF; 1: 发生 RXOVF。	R	0

### 8.6.7 SPI n中断清零寄存器 (SPI<sub>n</sub>\_IC) (n=0)

地址偏移量: 0x18

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	TXFIFOTHIC	0: 没有影响; 1: 清 TXFIFOTHIC 位。	W	0
2	RXFIFOTHIC	0: 没有影响; 1: 清 RXFIFOTHIC 位。	W	0
1	RXTOIC	0: 没有影响; 1: 清 RXTOIF 位。	W	0
0	RXOVFIC	0: 没有影响; 1: 清 RXOVFIF 位。	W	0

### 8.6.8 SPI n数据寄存器 (SPIn\_DATA) (n=0)

地址偏移量: 0x1C

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	DATA[15:0]	<p>写</p> <p>当 SPIn_STAT 寄存器 TX_FULL=0 (TX FIFO 非全满), SW 写入需要发送的数据帧到此寄存器。若 TX FIFO 为空, 且 SPI 控制器在总线上也并不繁忙, 传送数据将立刻发送。否则需等待前面的数据已经发送 (接收), 写入的数据才会被发送</p> <p>读</p> <p>当 SPIn_STAT 寄存器的 RX_EMPTY=0 (RX FIFO 非空) 时, SW 从此寄存器中读取数据。SW 读取此寄存器的数据时, SPI 控制器从 RX FIFO 的最近的帧中返回数据。若数据长度小于 16 位, 这些数据向右靠齐, 剩下的高位则填满 0。</p>	R/W	0

### 8.6.9 SPI n数据Fetch寄存器 (SPIn\_DF) (n=0)

地址偏移量: 0x20

Bit	Name	Description	Attribute	Reset
31:1	Reserved		R	0
0	DF	<p>SPI 数据 fetch 控制位。</p> <p>0: 禁止;</p> <p>1: 使能。</p>	R/W	0

# 9

## I2C

### 9.1 概述

I2C 总线是两线的双向通讯总线：串行时钟线（SCL）和串行数据线（SDA）。每个设备都通过唯一的地址进行识别，可以作为单一接收设备（如 LCD 驱动）或者兼具收发功能的发送设备（如存储器）。根据是否已经初始化数据传送或者分配设备地址，发送设备或接收设备可工作在主机或者从机模式下。I2C 允许多个主机同时存在于总线上，并且兼容 SMBus 2.0。

根据方向位（R/W）的状态，可在 I2C 总线上处理 2 种数据传送。

- 数据由主机发送器发送到从机接收器：  
主机发送的第一个字节是从机地址，接下来是数据字节；从机在接收到每个字节后，都返回一个应答信号。
- 数据由从机发送器发送到主机接收器：  
主机发送第一个字节（从机地址），从机返回一个应答信号。接下来由从机发送数据字节到主机。主机在所有接收到的数据之后会返回一个应答信号，除了最后一个字节。在接收到最后一个字节之后，主机返回一个非应答信号。主机设备产生所有串行时钟脉冲和 START 以及 STOP 信号。一个发送终止以 STOP 信号为依据，或者以重复 START 信号为依据。因为重复 START 信号是下一个串行发送的前提，I2C 总线是不会释放。

I2C 接口是以字节为单位，有 4 种操作模式：

- 主机发送模式；
- 主机接收模式；
- 从机发送模式；
- 从机接收模式。

## 9.2 特性

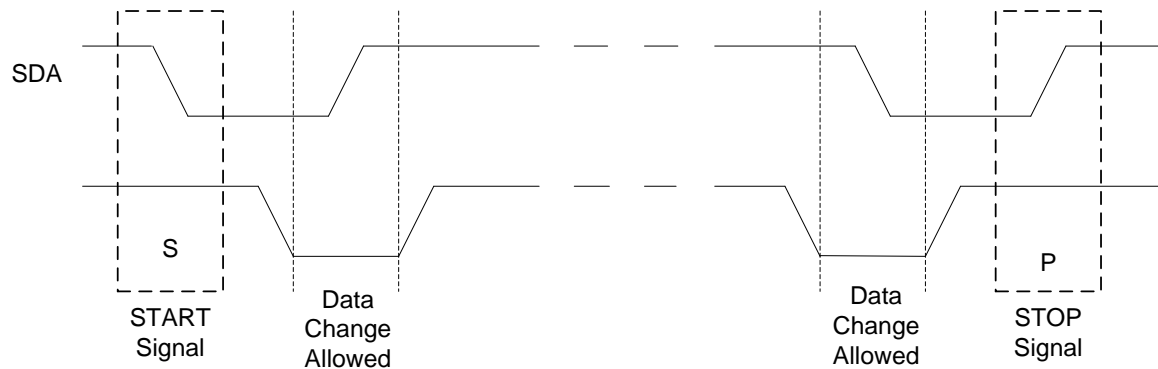
I2C 接口符合整个 I2C 规范，支持独自关闭 I2C 功能，而不影响同一 I2C 总线上的其它设备。

- 标准 I2C-compliant 总线接口可配置为主机或从机。
- I2C 主机特性：
  - 产生时钟；
  - 产生 START 和 STOP 信号
- I2C 从机特性：
  - 可编程控制 I2C 的地址检测；
  - 可选择识别多达 4 个不同的从机地址；
  - STOP 数据位检测。
- 支持不同的通讯速度：
  - 标准速度（高达 100KHz）；
  - 高速速度（高达 400KHz）。
- 仲裁负责管理多个同时发送的主机以避免总线上串行数据损坏。
- 可编程时钟允许调整的 I2C 发送速率。
- 主机和从机之间的数据传输是双向传送。
- 串行时钟同步允许设备使用不同的速率进行通信
- 串行时钟同步是被用于设备挂起和恢复串行传输的一种握手机制
- 监控模式用于监测所有 I2C 总线通讯，忽略从机地址。
- I2C 总线可用于进行测试和诊断。
- 7 位/10 位寻址的生成和检测以及通用调用。

## 9.3 引脚说明

Pin Name	Type	Description	GPIO Configuration
SCLn	I/O	I2C 串行时钟	开漏输出 输入时取决于 GPIO <sub>n</sub> _CFG
SDAn	I/O	I2C 串行数据	开漏输出 输入时取决于 GPIO <sub>n</sub> _CFG

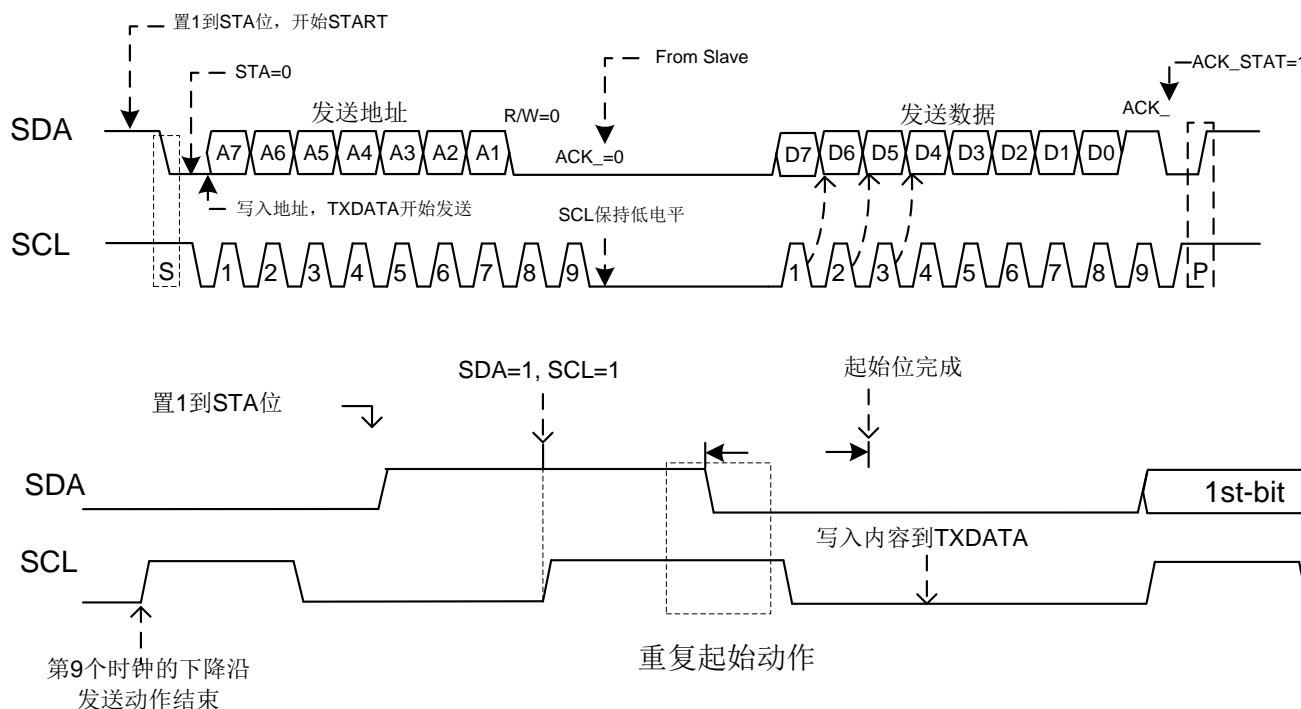
## 9.4 特性波形图



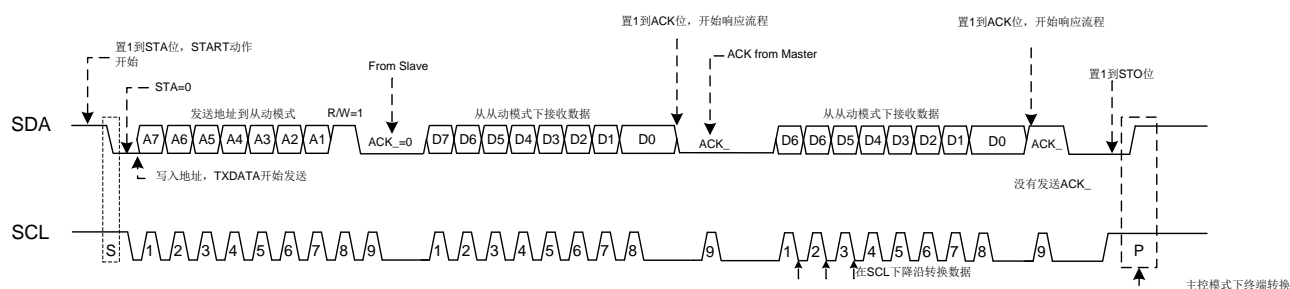


## 9.5 I2C主机模式

### 9.5.1 主机发送模式



### 9.5.2 主机接收模式



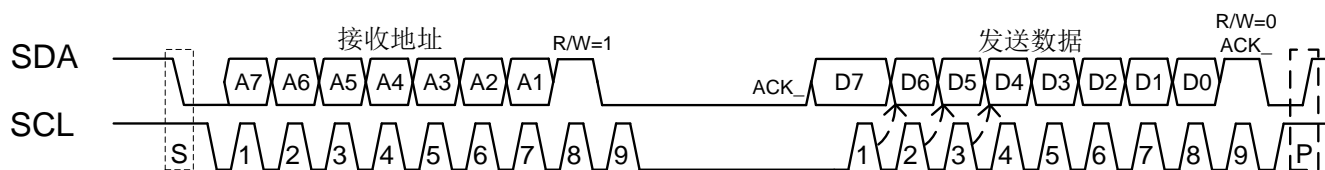
### 9.5.3 仲裁

在主机发送模式下, 仲裁逻辑检查每个发送的逻辑 1 在 I2C 总线上是否实际上表现为一个逻辑 1。若总线上的另一个设备否决了逻辑 1 并将 SDA 拉为低电平, 则仲裁丢失, I2C 模块立即从主机发送切换到从机接收。I2C 模块继续输出时钟脉冲 (SCL) 直至当前串行字节发送完成。

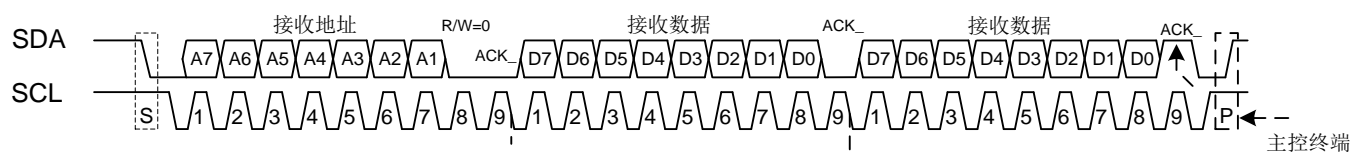
仲裁也可能在主机接收模式下丢失。在此模式下, 仲裁丢失仅仅发生在 I2C 模块返回一个 NAK 信号时。当总线上的另一个设备将信号拉为低电平时, 仲裁丢失。因为只有在串行字节的末端才会发生这种情况, I2C 模块不再产生时钟脉冲。

## 9.6 I2C从机模式

### 9.6.1 从机发送模式



### 9.6.2 从机接收模式



## 9.7 I2C寄存器

基地址: 0x4001 8000 (I2C0)

### 9.7.1 I2C n控制寄存器 (I2Cn\_CTRL) (n=0)

地址偏移量: 0x00

I2Cn\_CTRL 寄存器用于控制 I2C 接口的操作。

当 STA=1 时, 且 I2C 接口还未在主机模式时, 它进入主机模式, 检测总线的状态, 若总线空闲则产生一个 START 信号。若总线正在繁忙时, 则等待一个 STOP 信号 (释放总线), 在延迟半个内部时钟周期后产生一个 START 信号。如果 I2C 接口已经进入主机模式时, 且已经开始传送和接收数据, 它传送一个重复 START 信号。重复 START 信号可以在任何时候设置, 包括 I2C 接口处于编址的从机模式。

当在主机模式下 STO=1 时, 一个 STOP 信号发送到 I2C 总线上。当总线检测到 STOP 信号时, 自动将 STO 清零。从机模式下, 设置 STO 位可以从错误环境下恢复成正常状态。在这种情况下, 不会有 STOP 信号发送至总线。硬件就好像是已经接收到一个 STOP 信号, 然后切换到从机接收模式。

如果 STA 和 STO 都设置为 1 时, 若接口位于主机模式下, 一个 STOP 信号发送到 I2C 总线上, 然后再发送一个 START 信号。若 I2C 接口位于从机模式下, 产生一个内部 STOP 信号, 但不发送到总线上。

**\* 注:**

- 1、应该最后设置 I2CEN 位。
- 2、HW 将分配 SCL0/SCL1 和 SDA0/SDA1 引脚为开漏功能的输出引脚替代 GPIO 引脚。
- 3、接收数据时, ACK 和 NACK 位不能同时设置为 1。
- 4、主机模式下, 用户必须设置 ACK 或 NACK 位为 1, 以进行下一个 RX 处理。

Bit	Name	Description	Attribute	Reset
31:9	Reserved		R	0
8	I2CEN	I2C 接口使能位。 0: 禁止, STO 位被强制设为 0。 1: 使能, I2CEN 复位时, 总线状态处于丢失状态, 因此 I2CEN 不能用于暂时释放 I2C 总线, ACK 标志位可代替使用。	R/W	0
7	MODE	I2C 模式选择控制位。 0: 标准/高速模式; 1: 保留。		
6	Reserved		R	0
5	STA	START 位。 0: 无 START 信号或者产生重复的 START 信号; 1: 促使 I2C 接口进入主机模式, 发送一个 START 或者重复的 START 信号, 由 HW 自动清零。	R/W	0
4	STO	STOP 标志。 0: STOP 信号空闲; 1: 促使 I2C 接口在主机模式下发送一个 STOP 信号, 或者在从机模式下从错误的环境下恢复到正常的状态。由 HW 自动清零。	R/W	0
3	Reserved		R	0
2	ACK	ACK (SDA 为低电平) 信号标志。 0: 主机模式 → 无功能; 从机模式 → 接收地址或数据后返回一个 NACK; 1: 以下情况时, 在 SCLn 的应答时钟脉冲条件下产生一个应答信号: ■ 接收到从地址寄存器的地址; ■ 当 ADR 寄存器的通用调用位置 1 时, 接收到通用调用地址; ■ 当 I2C 处于主机接收模式时, 接收到数据字节; ■ 当 I2C 处于可寻址的从机接收模式时, 接收到数据字节。 硬件在发送完 ACK 后将自动清除此位。	R/W	0
1	NACK	NACK (SDA 为高电平) 信号标志。 0: 无功能。 1: 以下情况时, 在 SCL 的应答时钟脉冲条件下产生一个非应答信号: ■ 当 I2C 处于主机接收模式时, 数据字节被接收。 硬件在发送完 NACK 信号后将自动清除此位	R/W	0
0	Reserved		R	0

## 9.7.2 I2C n状态寄存器 (I2Cn\_STAT) (n=0)

地址偏移量: 0x04

发生 I2C 中断时, 检查该寄存器, 通过写 I2Cn\_CTRL 或 I2Cn\_TXDATA 寄存器, 所有的状态都自动清零。

I2CIF=1 时, SCL 串行时钟的低电平周期被延长, 串行发送则处于挂起状态。当 SCL 为高电平时, 则不受 I2CIF 状态的影响。

在 NVIC 中断控制器中使能 I2C 中断时, 下拉时间可以触发 I2C 中断:

- **START/Repeat START condition**
- **STOP condition**
- **定时器溢出**
- **发送或接收数据字节**
- **发送或接收 ACK**
- **发送或接收 NACK**

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15	I2CIF	I2C 中断标志位。 0: I2C 的状态未改变; 1: 读 → I2C 的状态改变, 写 → 将该标志位清零。	R/W	0
14:10	Reserved		R	0
9	TIMEOUT	超时状态位。 0: 未超时; 1: 超时。	R	0
8	LOST_ARB	仲裁丢失位。 0: 未仲裁丢失; 1: 仲裁丢失。	R	0
7	SLV_TX_HIT	0: 从机地址不匹配; 1: 从机地址匹配, 从机模式下为 TX 模式。	R	0
6	SLV_RX_HIT	0: 从机地址不匹配; 1: 从机地址匹配, 从机模式下为 RX 模式。	R	0
5	MST	主机/从机状态位。 0: I2C 处于从机状态; 1: I2C 处于主机状态。	R	0
4	START_DN	Start 处理状态位。 0: 无 START 信号; 1: 主机模式 → 发送一个 START 信号, 从机模式 → 接收到一个 START 信号。	R	0
3	STOP_DN	Stop 处理状态位。 0: 无 STOP 位; 1: 主机模式 → 发送一个 STOP 信号, 从机模式 → 接收到一个 STOP 信号。	R	0
2	NACK_STAT	NACK 处理状态位。 0: 没有接收到 NACK; 1: 接收到 NACK。	R	0
1	ACK_STAT	ACK 处理状态位。 0: 没有接收到 ACK; 1: 接收到 ACK。	R	0
0	RX_DN	RX 处理状态位。 0: 没有发送 RX ACK/NACK; 1: 已经发送 8 位 RX ACK/NACK。	R	0

### 9.7.3 I2C n TX数据寄存器 (I2Cn\_TXDATA) (n=0)

地址偏移量: 0x08

该寄存器包含发送的数据。

主机 TX 模式下, CPU 写入该寄存器可触发 TX 功能, 从机 TX 模式下, CPU 必须在下一次 TX 前写入该寄存器。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DATA[7:0]	发送的数据。	R/W	0x00

### 9.7.4 I2C n RX数据寄存器 (I2Cn\_RXDATA) (n=0)

地址偏移量: 0x0C

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	DATA[7:0]	包含接收的数据, RX_DN=1 时读该寄存器。	R	0x00

### 9.7.5 I2C n从机地址 0 寄存器 (I2Cn\_SLVADDR0) (n=0)

地址偏移量: 0x10

仅用于从机模式下。主机模式下, 该寄存器没有影响。

若该寄存器包含 0x00, I2C 不会在总线上应答任何地址。复位时, 寄存器 ADDR0~ADDR3 被清零, 此时处于禁止状态。

Bit	Name	Description	Attribute	Reset
31	ADD_MODE	从机地址模式。 0: 7 位地址模式; 1: 10 位地址模式。	RW	0
30	GCEN	通用调用地址使能位。 0: 禁止; 1: 使能通用调用地址 (0x00)。	RW	0
29:10	Reserved		R	0
9:0	ADDR[9:0]	I2C 从机地址。 ADD_MODE=1 时, ADD[9:0]有效; ADD_MODE=0 时, ADD[7:1]有效。	R/W	0

### 9.7.6 I2C n从机地址 1~3 寄存器 (I2Cn\_SLVADDR1~3) (n=0)

地址偏移量: 0x14, 0x18, 0x1C

Bit	Name	Description	Attribute	Reset
31:10	Reserved		R	0
9:0	ADDR[9:0]	I2C 从机地址。 ADD_MODE=1 时, ADD[9:0]有效; ADD_MODE=0 时, ADD[7:1]有效。	R/W	0

### 9.7.7 I2C n SCL高电平时间寄存器 (I2Cn\_SCLHT) (n=0)

地址偏移量: 0x20

\* 注: I2C 位频率 =  $I2Cn\_PCLK / (I2Cn\_SCLHT + I2Cn\_SCLLT)$ 。

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	SCLH[7:0]	计算 SCL 高电平周期时间。 SCL 高电平周期时间 = $(SCLH+1) * I2C0\_PCLK$ 周期	R/W	0x04

### 9.7.8 I2C n SCL低电平时间寄存器 (I2Cn\_SCLLT) (n=0)

地址偏移量: 0x24

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:0	SCLL[7:0]	计算 SCL 低电平周期时间。 SCL 低电平周期时间 = $(SCLL+1) * I2C0\_PCLK$ 周期	R/W	0x04

### 9.7.9 I2C n超时控制寄存器 (I2Cn\_TOCTRL) (n=0)

地址偏移量: 0x2C

当主机/从机 SCL 保持低电平超过以下时间时, 会发生超时现象:

$TO * 32 * I2C0\_PCLK$  周期

发生 I2C 超时中断时, I2C 发送会返回到“IDLE”状态, 并发出一个 TO 中断以通知用户。这就意味着超时后由 HW 释放 SCL/SDA。主机模式下, 用户可以在超时中断后发出一个 STOP 信号。

通过写 I2Cn\_CTRL 或 I2Cn\_TXDATA 寄存器可以自动清除超时状态。

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	TO[15:0]	计数检测超时。 0: 禁止超时检测; 1: 超时周期时间 = $N * I2Cn\_PCLK$ 周期。	R/W	0x0

# 10 USB FS设备接口

## 10.1 概述

USB 做为 PC 与外围通信的一种接口标准，以其快速、双向、同步、低成本、热插拔等特点大大满足了 PC 平台未来发展的需要。SONiX USB 控制芯片可将诸如鼠标、键盘、摇杆和游戏垫等计算机外设的人机交互推向最优化。

### USB 遵循的规范:

- 遵循USB规范V2.0
- 支持1个全速USB设备地址
- 支持1个控制端点和4个可配置端点（用于中断/Bulk发送）
- 集成USB收发器
- 5V转成3.3V输出regulator，连接1.5K的内部上拉电阻到D+

## 10.2 特性

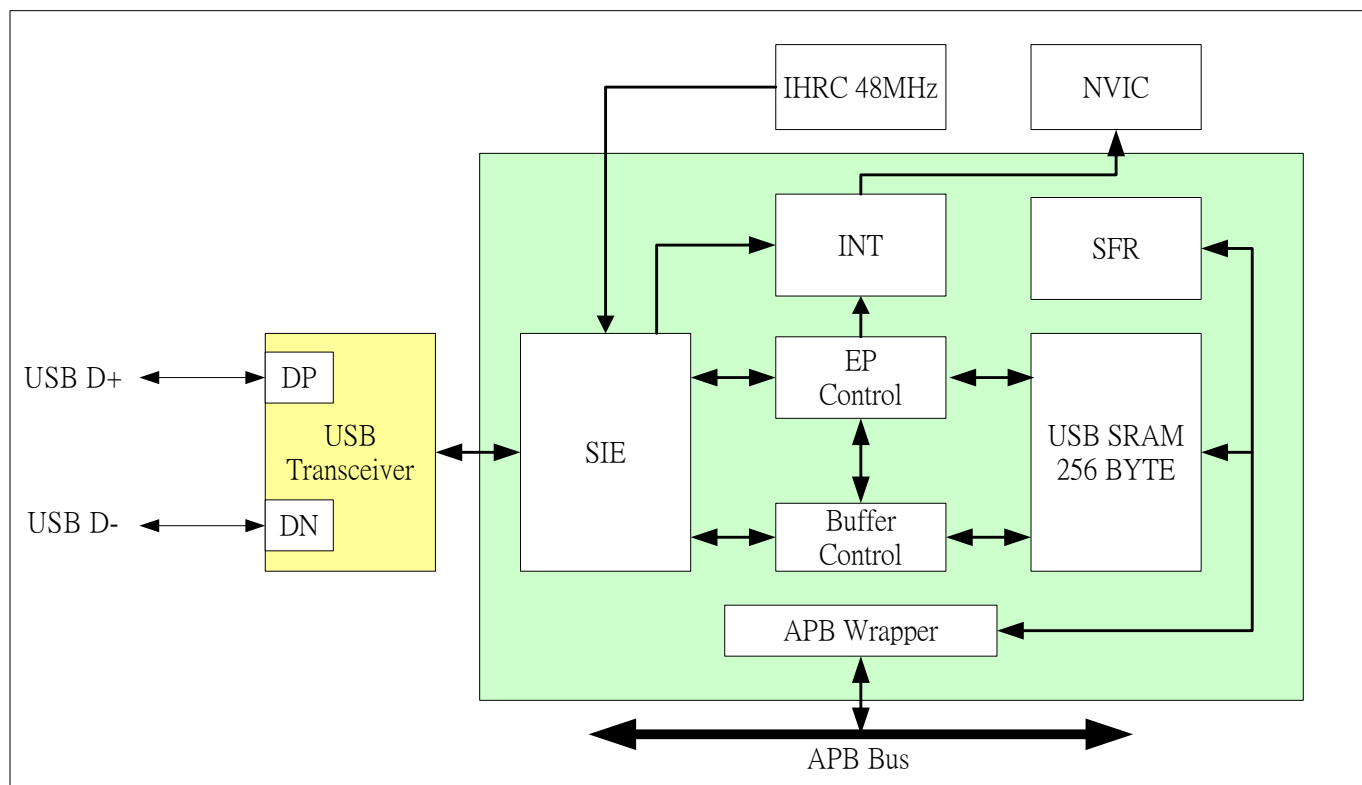
- 符合 USB 规范 V2.0
- 支持 1 个全速 USB 设备地址
- 支持 1 个控制端点，最大数据包尺寸为 8 字节，16 字节，32 字节或者 64 字节
- 支持 4 个可配置端点，用于中断/Bulk 发送
- 支持 USB SRAM 尺寸为 256 字节，与所有的 5 个端点共用
- 可灵活配置不同端点 FIFO 的地址偏移量（端点 0 除外）
- 5V 转 3.3V 的 regulator 上拉到内部 1.5K 电阻
- 集成 USB 收发器

系统时钟为 Fcpu/1、Fcpu/2、Fcpu/4 时，FS USB 正常工作

## 10.3 引脚说明

Pin Name	Type	Description	GPIO Configuration
DP	I/O	USB 差分信号 D+	N/A
DN	I/O	USB 差分信号 D-	N/A

## 10.4 框图



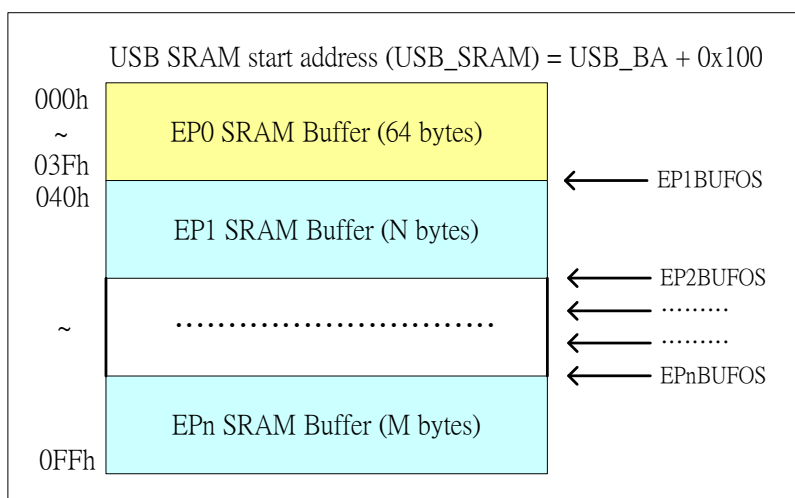
## 10.5 USB SRAM访问

USB SRAM 256 字节与 EP0~EP4, 共 5 个端点共用。用户应该在激活 USB 功能之前, 在缓存器中设置每个端点的有效起始地址。USB\_EPnBUFOS 模块可控制每个端点的有效起始地址。

访问 USB SRAM 的原则如下所示:

每个 EPnBUFOS 都设置为字对齐的, 2 LSB 位等于 0。

EPn SRAM 缓存器的最大长度由用户定义, 但每个端点都应该有自己的 EPn SRAM 缓存器, 且相互间不重叠。





## 10.6 USB机构

USB 机构实现单片机与 USB 主机的通信，硬件可独立地完成如下 USB 动作：

**USB 机构将完成：**

- 对接收到的数据译码，对将要发送的数据编码；
  - CRC 的校验和产生由硬件完成。若 CRC 错误，硬件将不给主机任何回应；
  - 硬件自动更新发送数据同步校准位；
  - USB 控制寄存器发送相应的 ACK/NAK/STALL 握手信号；
  - 不同类型令牌包（SETUP、IN、OUT）的识别。接收到有效令牌包后，对相应寄存器的状态位置“1”；
- 放置有效的接收数据到合适的端点 FIFO；
- 填充校验位；
  - 地址检查，丢掉无效地址的传输；
  - 端点检查，检查到来自主机的请求后，对寄存器的相关位进行设置。

**固件完成以下功能：**

- 通过接收 USB 设备请求上传相对应的列举数据；
- 填写/清空 FIFO。
- 复位/挂起/唤醒功能。
- 远程唤醒功能；
- USB 传输中，决定采用何种中断请求。

## 10.7 USB中断

USB 功能接收 USB 主机的命令并产生相关的中断，然后进入 USB\_IRQ\_Handler，此时要求固件检查 USB 的状态位以了解产生的是哪一种中断。

**在以下情况下会产生 USB 中断：**

- 端点 0 接收到设置令牌包（SETUP token）；
- 成功完成输入事务后，设备会接收到一个 ACK 应答信号。
- 端点在 ACK OUT 模式下，接收到数据后产生中断；
- USB 主机送出 USB 挂起请求；
- USB 复位；
- USB 处理完成后端点中断；
- 使能 NAK 中断时 NAK 握手。

## 10.8 USB枚举

典型的 USB 枚举流程如下：

- USB 主机发送 Setup 包后，再发送 DATA 包到地址 0，请求设备发送设备描述符。
- 固件收到请求，从 ROM 表中找到设备描述符。
- USB 主机发送输入控制时序，程序找到 USB 中的设备描述符通过内置 USB SRAM 回传给主机。
- USB 主机接收到描述符后，发送 SETUP 包和 DATA 包给地址 0 以给设备分配新 USB 地址。
- 无数据的状态控制阶段结束后，程序会将新地址存储在 USB 设备地址寄存器中。
- 主机利用新的 USB 地址请求发送设备描述符；
- 固件解码得到请求命令，并从程序存储列表找出设备描述符；
- 主机电脑发送读取命令，固件将设备描述符发送到 USB 总线上；
- 主机发出控制读命令，请求配置和说明描述符；
- 一旦设备接收到一个 Set Configuration 请求后，USB 功能开始使用；
- 程序要完成端点 0~N 的各交易操作。

## 10.9 USB寄存器

基地址: 0x4005 C000

R: 只读, W: 只写, R/W: 读/写

Register	Offset	R/W	Description	Reset Value
USB_INTEN	0x00	R/W	USB 中断使能寄存器	0x0000_0000
USB_INSTS	0x04	R	USB 中断事件状态寄存器	0x0000_0000
USB_INSTSC	0x08	W	USB 中断事件状态清零寄存器	0x0000_0000
USB_ADDR	0x0C	R/W	USB 设备地址寄存器	0x0000_0000
USB_CFG	0x10	R/W	USB 配置寄存器	0x0000_0000
USB_SGCTL	0x14	R/W	USB 信号控制寄存器	0x0000_0000
USB_EP0CTL	0x18	R/W	USB Endpoint 0 控制寄存器	0x0000_0000
USB_EP1CTL	0x1C	R/W	USB Endpoint 1 控制寄存器	0x0000_0000
USB_EP2CTL	0x20	R/W	USB Endpoint 2 控制寄存器	0x0000_0000
USB_EP3CTL	0x24	R/W	USB Endpoint 3 控制寄存器	0x0000_0000
USB_EP4CTL	0x28	R/W	USB Endpoint 4 控制寄存器	0x0000_0000
USB_EPTOGGLE	0x3C	R/W	USB Endpoint 数据 Toggle 寄存器	0x0000_003F
USB_EP1BUFOS	0x48	R/W	USB Endpoint 1 缓冲器偏移寄存器	0x0000_0000
USB_EP2BUFOS	0x4C	R/W	USB Endpoint 2 缓冲器偏移寄存器	0x0000_0000
USB_EP3BUFOS	0x50	R/W	USB Endpoint 3 缓冲器偏移寄存器	0x0000_0000
USB_EP4BUFOS	0x54	R/W	USB Endpoint 4 缓冲器偏移寄存器	0x0000_0000
USB_FRMNO	0x60	R	USB 帧数字寄存器	0x0000_0000
USB_PHYPRM	0x64	R/W	USB PHY 参数寄存器	0x0000_0000
USB_PHYPRM2	0x6C	R/W	USB PHY 参数寄存器 2	0x0000_0000
USB_PS2CTL	0x70	R/W	USB PS/2 控制寄存器	0x0000_0000
USB_RWADDR	0x78	R/W	USB FIFO 读/写地址寄存器	0x0000_0000
USB_RWDATA	0x7C	R/W	USB FIFO 数据寄存器	0x0000_0000
USB_RWSTATUS	0x80	R/W	USB FIFO 读/写状态寄存器	0x0000_0000
USB_RWADDR2	0x84	R/W	USB FIFO 读/写地址寄存器 2	0x0000_0000
USB_RWDATA2	0x88	R/W	USB FIFO 数据寄存器 2	0x0000_0000
USB_RWSTATUS2	0x8C	R/W	USB FIFO 读/写状态寄存器 2	0x0000_0000

### 10.9.1 USB中断使能寄存器 (USB\_INTEN)

地址偏移量: 0x00

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31	BUS_IE	总线事件中断使能位。 0: 禁止; 1: 使能, 包括 BUS_RESET、BUS_SUSPEND 和 BUS_RESUME 触发 USB 中断。	R/W	0
30	USB_SOF_IE	USB SOF 中断使能位。 0: 禁止; 1: 使能。	R/W	0
29	USB_IE	USB 事件中断使能位。 0: 禁止; 1: 使能, EP1~EP4 的 NAK 触发的 USB 中断除外。	R/W	0
28	BUSWK_IE	总线唤醒中断使能位。 0: 禁止; 1: 使能。		
27:5	Reserved		R	0
4	EPN_ACK_EN	EP1~4 ACK 中断使能位。 0: 禁止; 1: 使能。	R/W	0
3	EP4_NAK_EN	EP4 NAK 中断使能位。 0: 禁止; 1: 使能。	R/W	0
2	EP3_NAK_EN	EP3 NAK 中断使能位。 0: 禁止; 1: 使能。	R/W	0
1	EP2_NAK_EN	EP2 NAK 中断使能位。 0: 禁止; 1: 使能。	R/W	0
0	EP1_NAK_EN	EP1 NAK 中断使能位。 0: 禁止; 1: 使能。	R/W	0

## 10.9.2 USB中断事件状态寄存器（USB\_INSTS）

地址偏移量：0x04

复位值：0x0000 0000

Bit	Name	Description	Attribute	Reset
31	BUS_RESET	USB 总线复位信号标志位（>2.5us SE0）。 0：没有检测到总线复位信号； 1：检测到总线复位信号，写入 1 到 USB_INSTSC[31]清零。	R	0
30	BUS_SUSPEND	USB 总线挂起信号标志位（>3ms idle 状态）。 0：没有检测到总线挂起； 1：检测到总线挂起。	R	0
29	BUS_RESUME	USB 总线恢复信号标志位。 0：没有检测到总线恢复信号； 1：检测到挂起模式下的总线恢复信号，写入 1 到 USB_INSTSC[29]清零。	R	0
28:27	Reserved		R	0
26	USB_SOF	USB SOF 包接收标志位。 0：没有 USB SOF 包； 1：接收到 USB SOF 包，写入 1 到 USB_INSTSC[26]清零。	R	0
25	Reserved		R	0
24	EP0_PRESETUP	EP0 Setup token 包标志位。该标志位不触发 USB 中断。 0：没有 EP0 Setup token 包； 1：接收到 EP0 Setup token 包，写入 1 到 USB_INSTSC[24]清零。	R	0
23	EP0_SETUP	EP0 Setup transaction 标志位。 0：没有 EP0 Setup transaction； 1：完成 EP0 Setup transaction，写入 1 到 USB_INSTSC[23]清零。	R	0
22	EP0_IN	EP0 IN ACK transaction 标志位。 0：没有 EP0 IN ACK Transaction； 1：完成 EP0 IN ACK transaction，写入 1 到 USB_INSTSC[22]清零。	R	0
21	EP0_OUT	EP0 OUT ACK transaction 标志位。 0：没有 EP0 OUT ACK transaction； 1：完成 EP0 OUT ACK transaction，写入 1 到 USB_INSTSC[21]清零。	R	0
20	EP0_IN_STALL	EP0 IN STALL transaction 标志位。 0：没有 EP0 IN STALL transaction； 1：完成 EP0 IN STALL transaction，写入 1 到 USB_INSTSC[20]清零。	R	0
19	EP0_OUT_STALL	EP0 OUT STALL transaction 标志位。 0：没有 EP0 OUT STALL transaction； 1：完成 EP0 OUT STALL transaction，写入 1 到 USB_INSTSC[19]清零。	R	0
18	ERR_SETUP	接收到错误的 setup 数据标志位，该位不触发 USB 中断。 0：接收到普通的 8 字节 Setup DATA0； 1：Setup 数据不是 8 字节或 DATA0，写入 1 到 USB_INSTSC[18]清零。	R	0
17	ERR_TIMEOUT	超时状态标志位，该位不触发 USB 中断。 0：没有超时； 1：发送 IN 数据包后，主机 ACK 响应超时，写入 1 到 USB_INSTSC[17]清零。	R	0
16:12	Reserved		R	0
11	EP4_ACK	EP4 ACK transaction 标志位。 0：没有 EP4 ACK transaction； 1：完成 EP4 ACK transaction，写入 1 到 USB_INSTSC[11]清零。	R	0
10	EP3_ACK	EP3 ACK transaction 标志位。 0：没有 EP3 ACK transaction； 1：完成 EP3 ACK transaction，写入 1 到 USB_INSTSC[10]清零。	R	0
9	EP2_ACK	EP2 ACK transaction 标志位。 0：没有 EP2 ACK transaction； 1：完成 EP2 ACK transaction，写入 1 到 USB_INSTSC[9]清零。	R	0
8	EP1_ACK	EP1 ACK transaction 标志位。 0：没有 EP1 ACK transaction； 1：完成 EP1 ACK transaction，写入 1 到 USB_INSTSC[8]清零。	R	0
7:4	Reserved		R	0
3	EP4_NAK	EP4 NAK transaction 标志位。 0：没有 EP4 NAK transaction；	R	0

		1: 完成 EP4 NAK transaction, 写入 1 到 USB_INSTSC[3]清零。		
2	EP3_NAK	EP3 NAK transaction 标志位。 0: 没有 EP3 NAK transaction; 1: 完成 EP3 NAK transaction, 写入 1 到 USB_INSTSC[2]清零。	R	0
1	EP2_NAK	EP2 NAK transaction 标志位。 0: 没有 EP2 NAK transaction; 1: 完成 EP2 NAK transaction, 写入 1 到 USB_INSTSC[1]清零。	R	0
0	EP1_NAK	EP1 NAK transaction 标志位。 0: 没有 EP1 NAK transaction; 1: 完成 EP1 NAK transaction, 写入 1 到 USB_INSTSC[0]清零。	R	0

### 10.9.3 USB中断事件状态清零寄存器 (USB\_INSTSC)

地址偏移量: 0x08

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31	BUS_RESETC	0: 没有影响; 1: BUS_RESET 位清零。	W	0
30	Reserved		R	0
29	BUS_RESUMEC	0: 没有影响; 1: BUS_RESUME 位清零。	W	0
28:27	Reserved		R	0
26	USB_SOFC	0: 没有影响; 1: USB_SOF 位清零。	W	0
25	Reserved		R	0
24	EP0_PRESETUPC	0: 没有影响; 1: EP0_PRESETUP 位清零。	W	0
23	EP0_SETUPC	0: 没有影响; 1: EP0_SETUP 位清零。	W	0
22	EP0_INCC	0: 没有影响; 1: EP0_IN 位清零。	W	0
21	EP0_OUTC	0: 没有影响; 1: EP0_OUT 位清零。	W	0
20	EP0_IN_STALLC	0: 没有影响; 1: EP0_IN_STALL 位清零。	W	0
19	EP0_OUT_STALLC	0: 没有影响; 1: EP0_OUT_STALL 位清零。	W	0
18	ERR_SETUPC	0: 没有影响; 1: ERR_SETUP 位清零。	W	0
17	ERR_TIMEOUTC	0: 没有影响; 1: ERR_TIMEOUT 位清零。	W	0
16:12	Reserved		R	-
11	EP4_ACKC	0: 没有影响; 1: EP4_ACK 位清零。	W	0
10	EP3_ACKC	0: 没有影响; 1: EP3_ACK 位清零。	W	0
9	EP2_ACKC	0: 没有影响; 1: EP2_ACK 位清零。	W	0
8	EP1_ACKC	0: 没有影响; 1: EP1_ACK 位清零。	W	0
7:4	Reserved		R	-
3	EP4_NAKC	0: 没有影响; 1: EP4_NAK 位清零。	W	0
2	EP3_NAKC	0: 没有影响; 1: EP3_NAK 位清零。	W	0
1	EP2_NAKC	0: 没有影响; 1: EP2_NAK 位清零。	W	0
0	EP1_NAKC	0: 没有影响; 1: EP1_NAK 位清零。	W	0

### 10.9.4 USB设备地址寄存器（USB\_ADDR）

地址偏移量：0x0C

复位值：0x0000 0000

Bit	Name	Description	Attribute	Reset
31:7	Reserved		R	0
6:0	UADDR	USB 设备地址。	R/W	0

### 10.9.5 USB配置寄存器（USB\_CFG）

地址偏移量：0x10

复位值：0x0000 0000

Bit	Name	Description	Attribute	Reset
31	VREG33_EN	内部 VREG33 输出功能控制位，禁止 VREG33_EN 时，VREG33 切换为 IC_VDD。 0：禁止；1：使能。	R/W	0
30	PHY_EN	PHY 收发器功能控制位。进入睡眠/深度睡眠/深度掉电模式后，自动禁止 PHY。 0：禁止；1：使能。	R/W	0
29	DPPU_EN	内部 D+ 1.5k 上拉电阻功能控制位 0：禁止；1：使能。	R/W	0
28	SIE_EN	USB 串行接口 engine 使能控制位。 0：禁止；1：使能。	R/W	0
27	ESD_EN	USB ESD 保护使能控制位。 0：禁止；1：使能。	R/W	0
26	FLTDET_PDEN	D+和 D-之间的内部 175k 下拉电阻使能控制位。 0：禁止；1：使能。	R/W	0
25:4	Reserved		R	0
3	EP4_DIR	EP4 IN/OUT 方向设置位。 0：只与 IN token 包握手；1：只与 OUT token 包握手。	R/W	0
2	EP3_DIR	EP3 IN/OUT 方向设置位。 0：只与 IN token 包握手；1：只与 OUT token 包握手。	R/W	0
1	EP2_DIR	EP2 IN/OUT 方向设置位。 0：只与 IN token 包握手；1：只与 OUT token 包握手。	R/W	0
0	EP1_DIR	EP1 IN/OUT 方向设置位。 0：只与 IN token 包握手；1：只与 OUT token 包握手。	R/W	0

### 10.9.6 USB信号控制寄存器（USB\_SGCTL）

地址偏移量：0x14

复位值：0x0000 0000

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	BUS_DRVEN	驱动 USB 总线使能控制位。 0：没有驱动 USB 总线，写操作对 BUS_DP 和 BUS_DN 没有影响；1：驱动 USB 总线，通过设置 BUS_DP 和 BUS_DN 位来设置 D+/D-的状态。	R/W	0
1	BUS_DP	USB D+状态位。 0：D+为低电平； 1：D+为高电平。	R/W	0
0	BUS_DN	USB D-状态位。 0：D-为低电平； 1：D-为高电平。	R/W	0

### 10.9.7 USB EP0 控制寄存器 (USB\_EP0CTL)

地址偏移量: 0x18

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31	ENDP_EN	EP0 功能使能控制位。 0: 禁止, 与 EP0 SETUP/IN/OUT Token 没有握手; 1: 使能。	R/W	0
30:29	ENDP_STATE[1:0]	端点握手状态位。 00: NAK。 01: ACK。对于 IN transaction, 设备与 IN Transaction 握手 Data0/1; 对于 OUT transaction, 设备与 OUT token 和后面的 Data0/1 握手 ACK。 IN/OUT ACK Transaction 完成后, ENDP_STATE 自动返回到 NAK 状态。 10/11: INOUT_STALL。设备与 IN 或 OUT token 握手 STALL。USB Setup transaction 完成后, ENDP_STATE 自动返回到 NAK。	R/W	00
28	IN_STALL_EN	EP0 与 EP0 IN transaction 握手 STALL 使能控制位。 0: 禁止。 1: 使能。IN_STALL_EN 只在 EP0 IN token 使能有效。EP0 与 EP0 OUT transaction 的握手状态取决于 OUT_STALL_EN 和 ENDP_STATE 的设置。 USB Setup transaction 完成后, 该位自动清零。	R/W	0
27	OUT_STALL_EN	EP0 与 EP0 OUT transaction 握手 STALL 使能控制位。 0: 禁止; 1: 使能。OUT_STALL_EN 只在 EP0 OUT token 使能有效。EP0 与 EP0 IN transaction 的握手状态取决于 IN_STALL_EN 和 ENDP_STATE 的设置。 USB Setup transaction 完成后, 该位自动清零。	R/W	0
26:7	Reserved	-	R	0
6:0	ENDP_CNT[6:0]	端点字节计数位。 对于 IN transaction, ENDP_CNT 显示上传给主机的字节数。IN transaction 的最大字节数取决于 USB Device Descriptor 的 MaximumPacketSize0 宣告, 不能超过 64 字节。 对于 OUT transaction, ENDP_CNT 显示从主机接收到的字节数。	R/W	0

### 10.9.8 USB Epn控制寄存器 (USB\_EPnCTL, n = 1 ~ 4)

地址偏移量: 0x1C, 0x20, 0x24, 0x28

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31	ENDP_EN	EPn 功能使能控制位。 0: 禁止, 没有与 EPn IN/OUT token 握手; 1: 使能。	R/W	0
30:29	ENDP_STATE[1:0]	端点握手状态位。 00: NAK。对于 IN direction usage, 设备与 IN token 握手 NAK; 对于 OUT direction usage, 设备与 OUT token 握手 NAK。 01: ACK。对于 IN direction usage, 设备与 IN token 握手 Data0/1; 对于 OUT direction usage, 设备与 OUT token 和后面的 Data0/1 握手 ACK。 IN/OUT ACK Transaction 完成后, ENDP_STATE 自动返回到 NAK 状态。 10/11: STALL。对于 IN direction usage, 设备与 IN token 握手 STALL。对于 OUT direction usage, 设备与 OUT token 和后面的 Data0/1 握手 STALL。	R/W	0
28:7	Reserved	-	-	0
6:0	ENDP_CNT[6:0]	端点字节计数位。 对于 IN direction usage, ENDP_CNT 显示上传给主机的字节数。 对于 IN direction usage, ENDP_CNT 显示从主机接收到的字节数。	R/W	0



### 10.9.9 USB Epn数据Toggle寄存器 (USB\_EPTOGGLE)

地址偏移量: 0x3C

复位值: 0x0000 003F

Bit	Name	Description	Attribute	Reset
31:4	Reserved		R	0
3	EP4_DATA01	0: 设置 EP4 的 toggle 为 DATA0; 1: HW 自动设置 toggle 位。	R/W	1
2	EP3_DATA01	0: 设置 EP3 的 toggle 为 DATA0; 1: HW 自动设置 toggle 位。	R/W	1
1	EP2_DATA01	0: 设置 EP2 的 toggle 为 DATA0; 1: HW 自动设置 toggle 位。	R/W	1
0	EP1_DATA01	0: 设置 EP1 的 toggle 为 DATA0; 1: HW 自动设置 toggle 位。	R/W	1

### 10.9.10 USB Epn缓存器偏移寄存器 (USB\_EPnBUFOS, n = 1 ~ 4)

地址偏移量: 0x48, 0x4C, 0x50, 0x54

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:2	OFFSET[5:0]	每个端点数据缓存器的偏移地址位。有效的偏移地址为: USB_SRAM 地址 + {EPnBUFOS[7:2], 2'b00} USB_SRAM 地址 = USB_BA + 0x100 EP0 的偏移地址固定为 USB_SRAM 地址。	R/W	0
1:0	Reserved		R	0

### 10.9.11 USB帧数字寄存器 (USB\_FRMNO)

地址偏移量: 0x60

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:11	Reserved		R	0
10:0	FRAME_NO[10:0]	Start-Of-Frame (SOF) 包的 11 位帧数字。接收到 SOF 包时, H/W 自动更新该数字。	R	0

### 10.9.12 USB PHY参数寄存器 (USB\_PHYPRM)

地址偏移量: 0x64

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:26	PHY_PARAM[5:0]	USB PHY 参数值。建议设置为 0x20。	R/W	0
25:0	Reserved		R	0

### 10.9.13 USB PHY参数寄存器 2 (USB\_PHYPRM2)

地址偏移量: 0x6C

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:15	Reserved		R	0
14:0	PHY_PARAM2[14:0]	USB PHY 参数值。	R/W	0

### 10.9.14 USB PHY参数寄存器 (USB\_PS2CTL)

地址偏移量: 0x70

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31	PS2ENB	PS/2 内部 5K 上拉电阻控制位。	R/W	0
30:4	Reserved		R	0
3	SDA	PS/2 SDA 数据缓存器。	R/W	0
2	SCK	PS/2 SCK 数据缓存器。	R/W	0
1	SDAM	PS2/ SDA 模式控制位。	R/W	0
0	SCKM	SCKM PS/2 SCK 模式控制位。	R/W	0

### 10.9.15 USB R/W地址寄存器 (USB\_RWADDR)

地址偏移量: 0x78

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:2	RWADDR[5:0]	从 USB FIFO 读取地址或者写入地址到 USB FIFO。	R/W	0
1:0	Reserved		R	0

### 10.9.16 USB R/W数据寄存器 (USB\_RWDATA)

地址偏移量: 0x7C

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:0	RWDATA[31:0]	从 USB FIFO 读取数据或写入数据到 USB FIFO。	R/W	0

### 10.9.17 USB R/W状态寄存器 (USB\_RWSTATUS)

地址偏移量: 0x80

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1	R_STATUS	USB FIFO 读取状态。 *若 F/W 从 USB FIFO 中读取数据, 则该位置 1。 当硬件完成读操作后 (从 USB FIFO 读取的数据 RWDATA 写入到地址 RWADDR), 该位由硬件自动清零。	R/W	0
0	W_STATUS	USB FIFO 写入状态。 *若 F/W 写入数据到 USB FIFO, 则该位置 1。 当硬件完成写操作后 (RWDATA 作为读取的新数据, 然后写入到地址 RWADDR USB FIFO), 该位由硬件自动清零。	R/W	0



### 10.9.18 USB R/W地址寄存器 2 (USB\_RWADDR2)

地址偏移量: 0x84

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7:2	RWADDR[5:0]	从 USB FIFO 读取地址或者写入地址到 USB FIFO。	R/W	0
1:0	Reserved		R	0

### 10.9.19 USB R/W数据寄存器 2 (USB\_RWDATA2)

地址偏移量: 0x88

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:0	RWDATA[31:0]	从 USB FIFO 读取数据或写入数据到 USB FIFO。	R/W	0

### 10.9.20 USB R/W状态寄存器 2 (USB\_RWSTATUS2)

地址偏移量: 0x8C

复位值: 0x0000 0000

Bit	Name	Description	Attribute	Reset
31:2	Reserved		R	0
1	R_STATUS	USB FIFO 读取状态。 *若 F/W 从 USB FIFO 中读取数据，则该位置 1。 当硬件完成读操作后（从 USB FIFO 读取的数据 RWDATA 写入到地址 RWADDR），该位由硬件自动清零。	R/W	0
0	W_STATUS	USB FIFO 写入状态。 *若 F/W 写入数据到 USB FIFO，则该位置 1。 当硬件完成写操作后（RWDATA 作为读取的新数据，然后写入到地址 RWADDR USB FIFO），该位由硬件自动清零。	R/W	0

# 11 FLASH

## 11.1 概述

SONiX 32 位单片机集成了在线编程 FLASH 存储器的特性，以方便存储升级的代码。可以经由 SONiX 32 位单片机编程接口或者应用代码灵活地对 Flash 存储器进行编程控制。SONiX 32 位单片机为用户提供了安全选项以防止未经授权的不安全信息存储到 Flash 存储器中。

- 在编程或者擦除 Flash 过程中，单片机停止工作，即使外设（定时器、WDT、I/O、PWM 等）仍在正常工作。
- 应该在编写 Flash 或者擦除 Flash 之前，若看门狗处于使能状态，则需要将其清零。
- 擦除动作会将 Flash 页中的所有位都置为逻辑 0。
- HW 会控制时钟，并自动将 RAM 中的数据删除开始编程，编程完成后，HW 释放时钟，单片机会执行下一条指令。

## 11.2 嵌入式FLASH存储器

Flash 存储器为 32 位宽度的存储器，可用于存储代码和数据常量，位于芯片内存的特定地址。

内置的高性能 Flash 存储器模块有以下主要特性：存储器构架，Flash 存储器包括用户 ROM。

用户 ROM	32K × 8 位，分为 512 页，每页 64 字节
--------	-----------------------------

基于 AHB 协议，Flash 接口实现指令和数据的访问，实现了对 Flash 存储器的操作（编程/擦除），编程/擦除的操作可在产品的整个工作电压范围内执行。

## 11.3 特性

- 读取接口（32 位）
  - Flash 编程/擦除动作
  - 编译选项（Code Option）包括代码加密选项（CS）
- 嵌入式 Flash 存储控制器（FMC）管理主存储块和编译选项的执行。编程/擦除操作需要的高压由内部产生。主要的 Flash 存储器可以通过设定不同的代码加密级别（CS）进行读/写保护。
- 在对 Flash 存储器进行写选择时，任何试图对 Flash 存储器进行读取的操作都会中断总线，写操作完成后，才会开始正确的读操作。这就意味着在进行写/擦除操作时，不能安排读取代码或者数据。
- 在 Flash 存储器进行写和擦除操作时，IHRC 应该由 FMC 设为 ON 状态。可以通过 ICP 和 ISP 对 Flash 存储器进行编程和擦除操作。

## 11.4 机构

Block		Name	Base Address	Size (Byte)
SN32F268 ROM 32KB	User ROM 30KB	Page 0	0x00000000 ~ 0x0000003F	64
		Page 1	0x00000040 ~ 0x0000007F	64
		.	.	.
		.	.	.
		.	.	.
		Page 478	0x00007780 ~ 0x000077BF	64
		Page 479	0x000077C0 ~ 0x000077FF	64
	Boot Loader 2KB	Page 480	0x00007800 ~ 0x0000783F	64
		.	.	.
		.	.	.
		.	.	.
		Page 511	0x00007FC0 ~ 0x00007FFF	64

## 11.5 读操作

作为一个通用的存储空间，嵌入式 Flash 模块可以直接进行访问。读操作是访问 Flash 模块内容并提供需要的数据。

读取界面由一个读取控制器（一面访问 Flash 存储器），和一个 AHB 接口（另一面与 CPU 联系）组成。读取界面的主要任务是产生控制信号，从 Flash 存储器中读取 CPU 需要的内容。

## 11.6 编程/擦除

Flash 存储器的擦除操作按页执行。

为了保证不产生过度的编程，IHRC 提供 Flash 编程和擦除时钟。

## 11.7 嵌入式引导加载程序

嵌入式引导加载程序利用 USB 接口来重新编程 Flash 存储器。

## 11.8 FLASH存储控制器（FMC）

FMC 控制 Flash 存储器的编程和擦除。

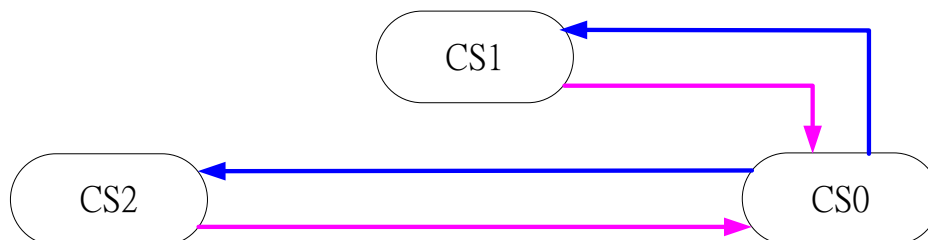
### 11.8.1 代码加密（CS）

代码加密是一种允许用户使能不同加密等级的机制，以便访问内置 Flash 和限制 ISP 操作。

\* 注：只有 MCU 重新启动后，改变的代码加密才会生效。

User ROM		CS0	CS1	CS2	Description
WRITER	Read	O	X	X	
	Erase	O	O(*)	O(*)	(*)若 CS 级别不是 CS0，则在擦除后会变成 CS0。
	Program	O	O	O	
FW (Flash emulation)	Read	O	O	O	
	Erase	O	O	O	
	Program	O	O	O	
SWD	Read	O	X	X	
	Erase	O	X	X	
	Program	O	X	X	

\* 注：用户可能试图改变安全级别，从 CS2 改为 CS0，或者从 CS1 改为 CS0，HW 会有以下操作：  
1、首先将用户 ROM 区域擦除，用户不能在调试模式下执行该操作，这是由于 SWD 通信在擦除操作时可能会失败。  
2、更新安全级别。



— includes:  
- New option byte programming

— includes:  
- Option byte erase  
- Mass Erase

## 11.8.2 编程FLASH存储器

Flash 存储器可以一次编程 1 页（64 字节），通过执行标准的页写入操作，CPU 可以编程主要的 Flash 存储器。FLASH\_CTRL 寄存器的 PG 位必须设置为 1。写入数据到 FLASH\_DATA 寄存器时，FMC 准备增加数据地址，检查地址是否已经完成编程，若发生下列错误，跳过编程操作，通过 FLASH\_STATUS 寄存器的 PGERR 位发出一个警告。

开始擦除/编程，发现地址已经超出页边界；

开始擦除/编程，发现地址不符合规则（>ROM 尺寸）；

填入数据和地址，已经超出也边界。

标准模式下的主要 Flash 存储器编程流程如下：

- 1、设置 FLASH\_CTRL 寄存器的 PG 位为 1；
- 2、在 FLASH\_ADDR 寄存器中指定地址；
- 3、等待 BUSY 位复位；
- 4、写入连续的数据直到所有数据都填入 FLASH\_DATA 寄存器中；
- 5、等待 BUSY 位复位；
- 6、设置 START 位开始编程；
- 7、等待 BUSY 位复位；
- 8、（可选）读取编程的值并进行校验。

## 11.8.3 擦除操作

Flash 存储器可以按页擦除，也可全部擦除（批量擦除）。

### 11.8.3.1 按页擦除

可以利用 FMC 的页擦除特性将 Flash 存储器的整页进行擦除。要擦除整页，其流程如下：

- 1、设置 FLASH\_CTRL 寄存器的 PER 位为 1；
- 2、对 FLASH\_ADDR 寄存器进行编程来选择一页进行擦除；
- 3、设置 FLASH\_CTRL 寄存器的 SATRTE 位为 1；
- 4、等待 BUSY 位复位；
- 5、（可选）读取擦除页的内容并进行校验。

### 11.8.3.2 批量擦除

当 Flash 存储器读保护选项由受保护变为不受保护时，可以由 HW 在重新烧录读保护选项前将用户 ROM 进行批量擦除。

## 11.9 读保护

在编译选项中设置代码安全字节可以激活读保护选项。

当 Flash 存储器读保护选项由受保护变为不受保护时，可以由 HW 在重新烧录读保护选项前将用户 ROM 进行批量擦除。

## 11.10 HW CHECKSUM

HW checksum 是用户 ROM 的 checksum，若使能读保护，用户可以通过烧录器或者 ISP AP 读出 HW 的 checksum 值。

## 11.11 FMC寄存器

基地址：0x4006 2000

### 11.11.1 Flash低电压控制寄存器（FLASH\_LPCTRL）

地址偏移量：0x00

Bit	Name	Description	Attribute	Reset
31:16	FMCKEY	FMC verify 位。 读取为 0，需要写入数据到该寄存器时，必须写入 0x5AFA 到 FMCKEY，否则会忽略对该寄存器的写动作。	W	0
15:4	Reserved		R	0
3:0	LPMODE[3:0]	Flash 低电压模式选择位。 0000b: HCLK < 24MHz; 0101b: HCLK ≥ 24MHz; 其他：保留，可能会引起意外的错误而强制 MCU 进入 hardfault Handler。	R/W	0

### 11.11.2 Flash状态寄存器（FLASH\_STATUS）

地址偏移量：0x04

复位值：0x0000 0000

Bit	Name	Description	Attribute	Reset
31:3	Reserved		R	0
2	ERR	编程错误标志位。 0: 读 → 无错误，写 → 将该位清零； 1: 当开始擦除/编程，发现地址已经超出页边界； 开始擦除/编程，发现地址不符合规则（>ROM 尺寸）； 填入数据和地址，已经超出页边界。时该位由硬件置 1。	R/W	0
1	Reserved		R	0
0	BUSY	繁忙指示标志位。 0: Flash 操作不繁忙； 1: 正在处理 Flash 操作，在 Flash 操作开始时设置该位（同时 EOP 位清零），操作完成后或者 HW 出错时复位。	R	0

### 11.11.3 Flash控制寄存器（FLASH\_CTRL）

地址偏移量：0x08

Bit	Name	Description	Attribute	Reset
31:8	Reserved		R	0
7	CHK	Checksum 计算选择。 该位只能由 SW 设置为 1，BUSY 位复位时，该位也复位。	R/W	0
6	STARTE	擦除/编程操作起始位。 1: 触发擦除/编程操作后，只能由 SW 将该位置 1，BUSY 位复位时，该位也复位； 0: 停止/完成擦除/编程操作。	R/W	0
5:2	Reserved		R	0
1	PER	页擦除操作选择位。 该位只能由 SW 设置为 1，BUSY 位复位时，该位也复位。	R/W	0
0	PG	Flash 编程操作选择位。 该位只能由 SW 设置为 1，BUSY 位复位时，该位也复位。	R/W	0

### 11.11.4 Flash数据寄存器（FLASH\_DATA）

地址偏移量：0x0C

进行页编程操作时，应该由 SW 操作该寄存器，显示需要编程的数据。

Bit	Name	Description	Attribute	Reset
31:0	DATA[31:0]	需要编程的数据。	R/W	0

### 11.11.5 Flash地址寄存器（FLASH\_ADDR）

地址偏移量：0x10

必须由 SW 升级需要擦除或者编程的 Flash 地址，填满 Flash 地址之前，PG 位或者 PER 位需设置为 1。

\* 注：当 FLASH\_STATUS 寄存器的 BUSY 位置 1 时，会阻止对该寄存器的写访问。

Bit	Name	Description	Attribute	Reset
31:0	FAR[31:0]	Flash 地址。 选择按页擦除时，则选择一个页面进行擦除；或者选择按页编程时，则选择一个页面进行编程。	R/W	0

### 11.11.6 Flash Checksum寄存器（FLASH\_CHKSUM）

地址偏移量：0x14

Bit	Name	Description	Attribute	Reset
31:16	Reserved		R	0
15:0	CHKSUM[15:0]	用户 ROM 的 Checksum 值。	R	0

# 12 SERIAL-WIRE调试（SWD）

## 12.1 概述

SWD 功能集成到 ARM Cortex-M0 中，配置好的 ARM Cortex-M0 可支持 4 个断点和 2 个监控点。

## 12.2 特性

- 支持 ARM Serial Wire 调试（SWD）模式；
- 直接调试访问所有存储器、寄存器和外设；
- 调试期不需要目标源；
- 4 个断点；
- 2 个可作为触发器的数据观察点。

## 12.3 引脚说明

Pin Name	Type	Description	GPIO Configuration
SWCLK	I	SWD 模式下的 Serial Wire 时钟引脚。	
SWDIO	I/O	SWD 模式下的 Serial Wire 数据输入/输出引脚。	

## 12.4 调试注意事项

### 12.4.1 局限性

调试模式改变了 ARM Cortex-M0 CPU 用于降低功耗的工作模式，而这一影响还会波及到整个系统。这意味着不应该在调试阶段进行功耗测量，否则测量结果会比正常应用操作时偏高。

在调试阶段，每当 CPU 停止工作时，SysTick 定时器会自动停止工作，其它的外设则不受影响。

### 12.4.2 恢复调试功能

用户代码可以禁止 SWD 功能以便将 P3.6 和 P3.7 作为 GPIO 使用，此时不能通过 SWD 功能来调试或者下载 FW。

在引导程序期间，SONiX 提供引导加载程序来检查 P3.5（引导引脚）的状态，若 P3.5 为低电平，MCU 会用引导加载程序代替用户程序，故不能禁止 SWD 功能。

退出引导加载程序后，用户程序可以将 P3.5 作为其它功能（如 GPIO）使用。

\* 注：我们强烈建议不要将 BOOT 引脚设置为输出引脚用于驱动 LED，否则 BOOT 引脚的状态会在 BOOT 步骤时变低。

### 12.4.3 SWD引脚上的内部上拉/下拉电阻

为了避免不受控制的 I/O 电平，在 SWD 输入引脚上内置了上拉电阻和下拉电阻。

- SWDIO/JTMS：内部上拉；
- SWCLK/JTCK：内部下拉。

SW 一旦禁止了 SWD 功能，GPIO 控制器将再次控制相关功能。

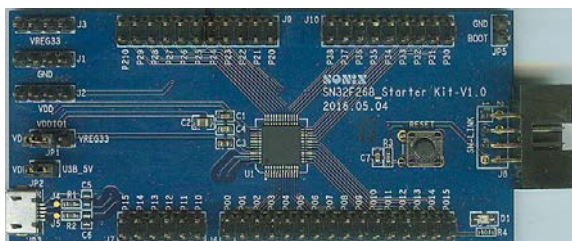


# 13 开发工具

SONiX 提供一套嵌入式的 ICE 仿真系统进行 32 位系列单片机固件开发。

SONiX 32 位系列嵌入式 ICE 仿真系统包括：

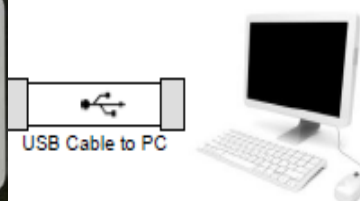
- SONiX 32 位 MCU Starter-Kit.
- SN-LINK-V2/V3
- USB 线，用于 SN-LINK-V2/V3 和 PC 的通讯
- IDE (KEIL RVMDK)



SONiX 32-bit MCU Starter-Kit.



SN-LINK-V2



IDE Tools

SONiX 32 位系列的嵌入式 ICE 仿真特性如下：

- 目标板的工作电压：2.5V~5.5V
- 高达 4 个硬件断点
- 系统时钟速率高达 48MHz
- 振荡器支持 IHRC, ILRC

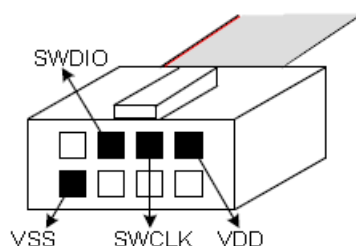
SONiX 32 位系统的嵌入式 ICE 仿真时有下拉限制：

- SWCLK 和 SWDIO 引脚与 GPIO 引脚共用，在嵌入式 ICE 模式下，该硬件的 GPIO 功能被屏蔽。

## 13.1 SN-LINK-V2/V3

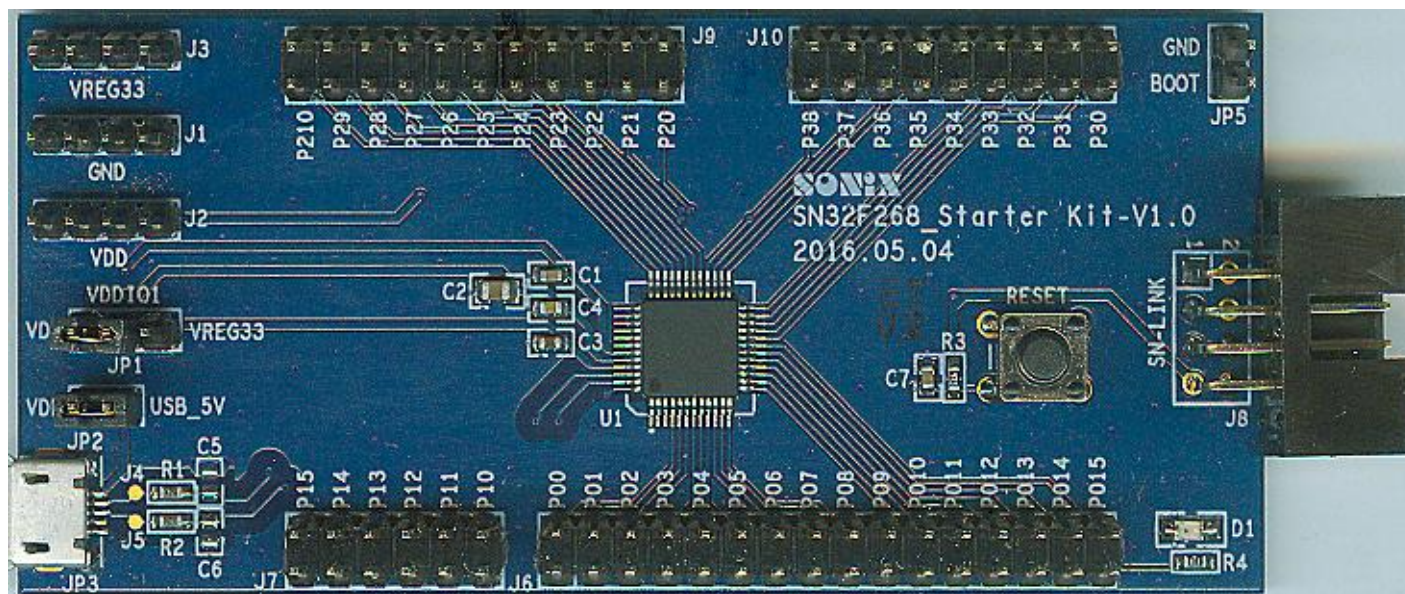
SN-LINK-V2/V3 是一个高速仿真器，用于 SONiX 32 位系列单片机的仿真，基于 SWD 协议进行调试和编程。除了调试功能外，程序员还可以通过 SN-LINK-V2/V3 从 PC 下载固件到单片机进行设计生产，甚至是大批量的生产。

SN-LINK-V2/V3 通过 SWD 接口连接到 SONiX 32 位 MCU，该模块的引脚定义如下图所示：



## 13.2 SN32F268 STARTER-KIT

SN32F268 Starter-kit 是一个简易的开发平台，包括 SN32F268 实际芯片和 I/O 接口，用于输入信号或驱动用户应用的额外装置，当目标板未准备好时，它也是一个简单的开发平台。由于 SN32F268 系列单片机集成 SWD 调试电路，目标板也可替换 Starter-kit。



- JP3: Micro USB 接口。
- JP2: USB 电源接口。
- JP1: VDDIO1 电源接口，选择 VDDIO1 (P1.0~P1.5) (板上 5.0V/3.3V) 电源。
- J1: GND 接口。
- J2: VDD 接口。
- J3: VREG33 输出接口。
- U1: SN32F268BF 实际 IC。
- RESET button: 外部复位触发源。
- D1: 测试 LED。
- J8: SN-LINK 接口。
- JP5: 短路以前置 MCU 位于 Boot loader。

# 14 电气特性

## 14.1 极限参数

Supply voltage (Vdd).....	- 0.3V ~ 5.5V
Input in voltage (Vin).....	Vss – 0.2V ~ Vdd + 0.2V
Operating ambient temperature (Topr).....	-40°C ~ + 85°C
Storage ambient temperature (Tstor) .....	-40°C ~ + 125°C

## 14.2 电气特性

All of voltages refer to Vss, Typical Vdd = 5.0V, Fosc = 12MHz, ambient temperature is 25℃ unless otherwise note.							
PARAMETER	SYM.	DESCRIPTION		MIN.	TYP.	MAX.	UNIT
Operating Voltage	Vdd1	Supply voltage for core and external rail		2.5	5.0	5.5	V
	Vdd2	USB mode		3.1	5.0	5.25	V
VDD rise rate	VPOR	VDD rise rate to ensure internal power-on reset		0.05	-	-	V/ms
Power Consumption							
Supply Current	Idd1	Normal mode	System clock = 48MHz [1][2][3]	-	12	-	mA
	Idd2	Sleep Mode	System clock = 32KHz [1][3][4]	-	160	230	uA
	Idd3	Deep-sleep Mode	Vdd=5V [1][5]	-	5	16	uA
Port Pins, RESET pin							
High-level input voltage	VIH			0.7Vdd	-	Vdd	V
Low-level input voltage	VIL			Vss	-	0.3Vd d	V
Input voltage	Vi			0	-	Vdd	V
Output voltage	Vo			0	-	Vdd	V
I/O port pull-up resistor	RPU	Vin = Vss , Vdd = 5.0V		30	50	70	KΩ
I/O port pull-down resistor (SWD pull-down pin)	RPD	Vin = 5.0V		30	50	70	KΩ
I/O High-level output source current	IOH	VOP = Vdd – 0.5V;		6	10	-	mA
I/O Low-level output sink current	IOL	VOP = Vss + 0.5V		12	20	-	mA
FLASH							
Endurance time	TEN	Erase + Program		10K	*100K	-	Cycle
Page erase time	TME	All User ROM memory.		-	5	-	ms
Page Programming time	TPG	1 -Page (64 bytes).		-	5	-	ms
MISC							
Low Voltage Detector	LVD	Interrupt/Reset	LVD24	2.2	2.4	2.6	V
			LVD33	3.1	3.3	3.5	V
3.3V Regulator Output voltage	Vreg33	VCC ≥ 3.60V, IVREG33 >= 60 mA		3.03	-	3.27	V
IHRC Freq.	FIHRC	T=25℃, Vdd=5V		11.97	12	12.03	MHz
		T=-40℃~85℃, Vdd=5V		11.64	12	12.36	MHz
	FIHRC 2	T=-40℃~85℃, Vdd=5V, USB function ON		11.97	12	12.03	MHz

\* Parameters with star mark are non-verified design reference.

[1] IDD measurements were performed with all pins configured as GPIO outputs driven LOW and pull-up resistors disabled and VDD=5V.

[2] IHRC and ILRC are enabled.

[3] LVD and GPIO peripherals are enabled.

[4] IHRC is disabled, ILRC is enabled.

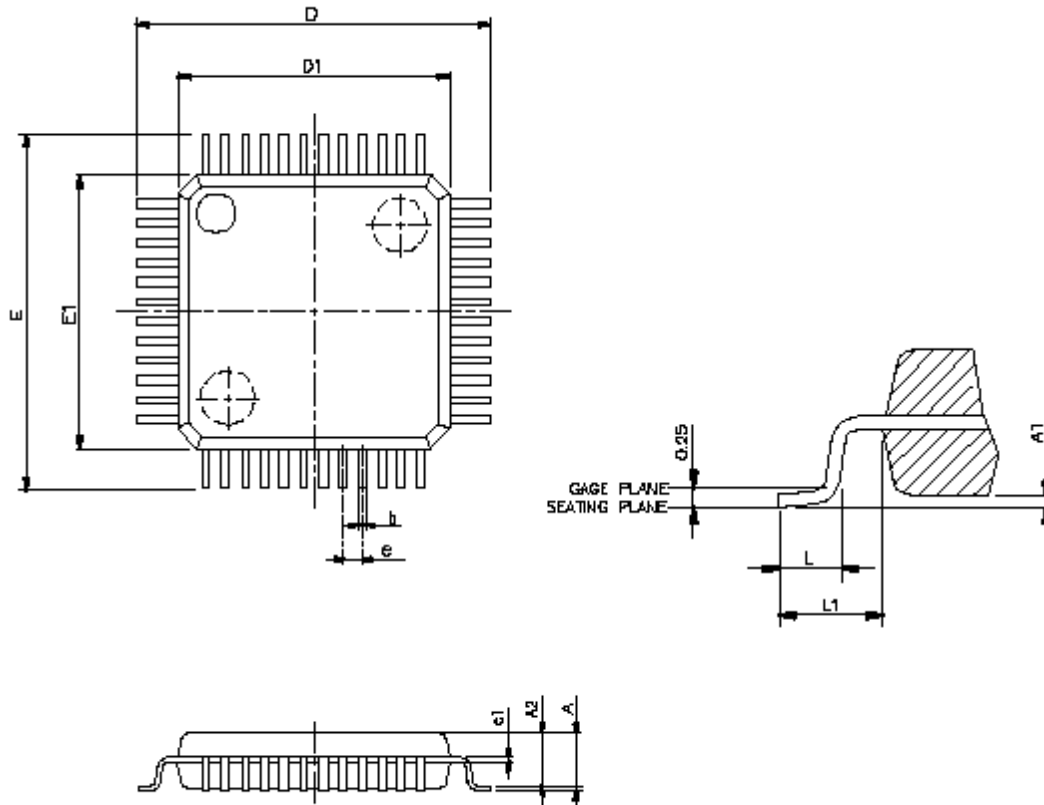
[5] All oscillators and analog blocks are turned off.

## 15 FLASH ROM烧录引脚

SN32F260 系列烧录信息													
单片机名称		SN32F268F		SN32F267J		SN32F265J		SN32F2641J		SN32F264S/X		SN32F263X	
MP PRO Writer		Flash IC / JP3 引脚配置											
Number	Name	Number	Pin	Number	Pin	Number	Pin	Number	Pin	Number	Pin	Number	Pin
1	VDD	44	VDD	41	VDD	31	VDD	26	VDD	2	VDD	1	VDD
2	GND	48	VSS	45	VSS	3	VSS	2	VSS	6	VSS	15	VSS
3	CLK	5	P0.1	4	P0.1	5	P0.1	4	P0.1	8	P0.1	6	P0.1
4	CE												
5	PGM	26	P3.6	25	P3.6	21	P3.6	18	P3.6	20	P3.6	16	P3.6
6	OE	27	P3.7	26	P3.7	22	P3.7	19	P3.7	21	P3.7	17	P3.7
7	D1												
8	D0												
9	D3												
10	D2												
11	D5												
12	D4												
13	D7												
14	D6												
15	VDD												
16	-												
17	HLS												
18	RST												
19	-												
20	ALSB/PDB	6	P0.2	5	P0.2	6	P0.2	5	P0.2	9	P0.2	7	P0.2

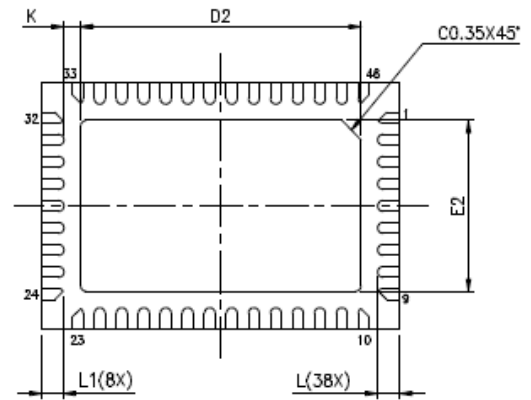
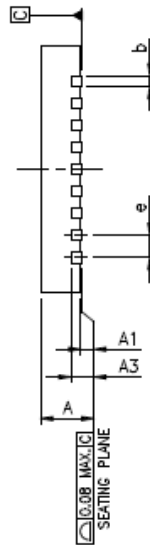
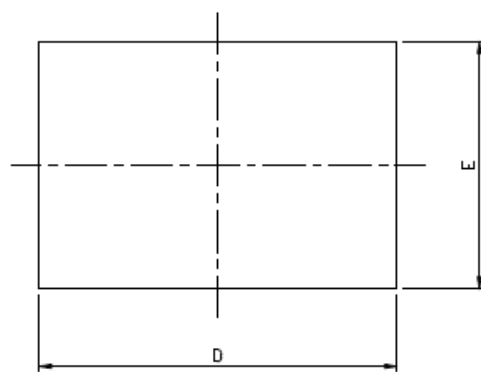
# 16 封装信息

## 16.1 LQFP 48 PIN



SYMBOLS	MIN	NOR	MAX
	(mm)		
A	-	-	1.6
A1	0.05	-	0.15
A2	1.35	-	1.45
c1	0.09	-	0.16
D	9.00 BSC		
D1	7.00 BSC		
E	9.00 BSC		
E1	7.00 BSC		
e	0.5 BSC		
b	0.17	-	0.27
L	0.45	-	0.75
L1	1 REF		

## 16.2 QFN 46 PIN



PAD SIZE : 213X13\* MIL

## NOTES :

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSION b APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15mm AND 0.30mm FROM THE TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION b SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
3. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.

JEDEC OUTLINE	PACKAGE TYPE		
	N/A		
SYMBOLS	MIN.	NOM.	MAX.
A	0.70	0.80	0.90
A1	0.00	0.02	0.05
A3	0.203 REF.		
b	0.15	0.20	0.25
D	6.50 BSC		
E	4.50 BSC		
e	0.40 BSC		
K	0.20	—	—

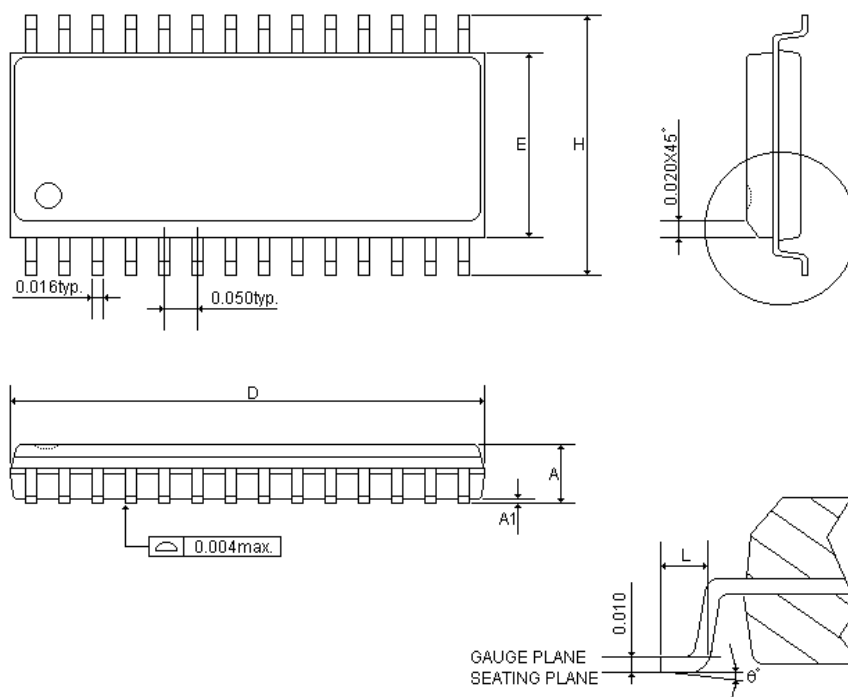
PAD SIZE	D2			E2			L			L1			LEAD FINISH		JEDEC CODE
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	Pure Tin	PPF	
212X13* MIL	5.05	5.10	5.15	3.05	3.10	3.15	0.35	0.40	0.45	0.33	0.38	0.43	V	X	N/A

△ "\*"表示汎用字元,此汎用字元可能被其它不同字元所取代,實際的字元請參照bonding diagram所示。

"\*" is an universal character, which means maybe replaced by specific character, the actual character please refers to the bonding diagram.

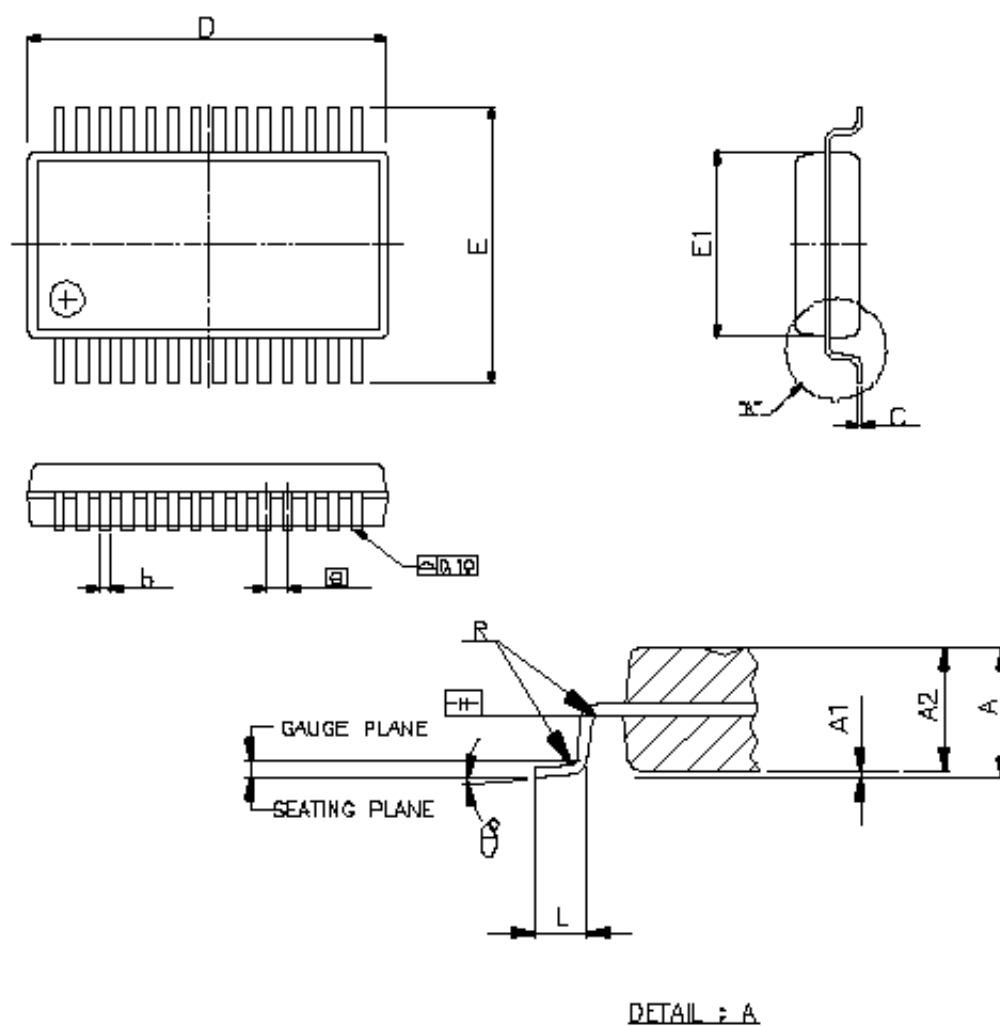


## 16.3 SOP 28 PIN



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	0.093	0.099	0.104	2.362	2.502	2.642
A1	0.004	0.008	0.012	0.102	0.203	0.305
D	0.697	0.705	0.713	17.704	17.907	18.110
E	0.291	0.295	0.299	7.391	7.493	7.595
H	0.394	0.407	0.419	10.008	10.325	10.643
L	0.016	0.033	0.050	0.406	0.838	1.270
θ°	0°	4°	8°	0°	4°	8°

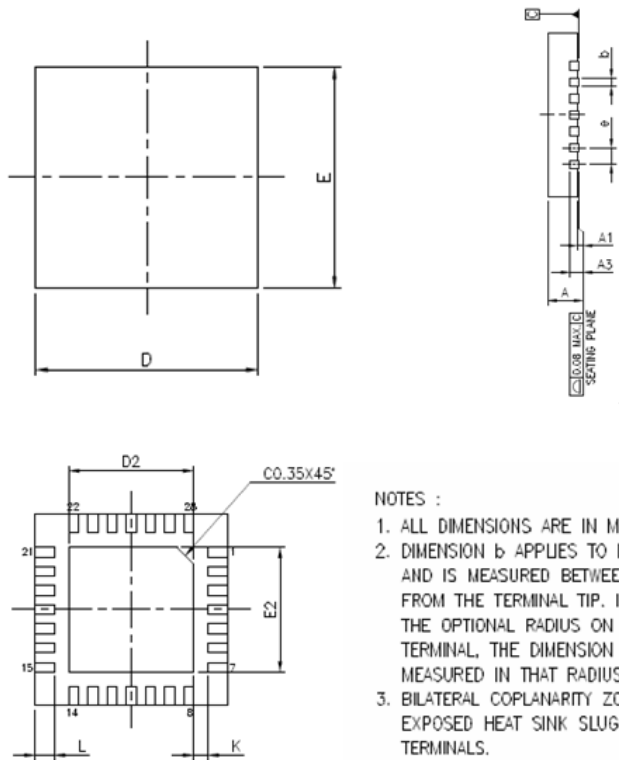
## 16.4 SSOP 28 PIN



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	-	-	0.08	-	-	2.13
A1	0.00	-	0.01	0.05	-	0.25
A2	0.06	0.07	0.07	1.63	1.75	1.88
b	0.01	-	0.01	0.22	-	0.38
C	0.00	-	0.01	0.09	-	0.20
D	0.39	0.40	0.41	9.90	10.20	10.50
E	0.29	0.31	0.32	7.40	7.80	8.20
E1	0.20	0.21	0.22	5.00	5.30	5.60
[e]	0.0259BSC			0.65BSC		
L	0.02	0.04	0.04	0.63	0.90	1.03
R	0.00	-	-	0.09	-	-
θ°	0°	4°	8°	0°	4°	8°



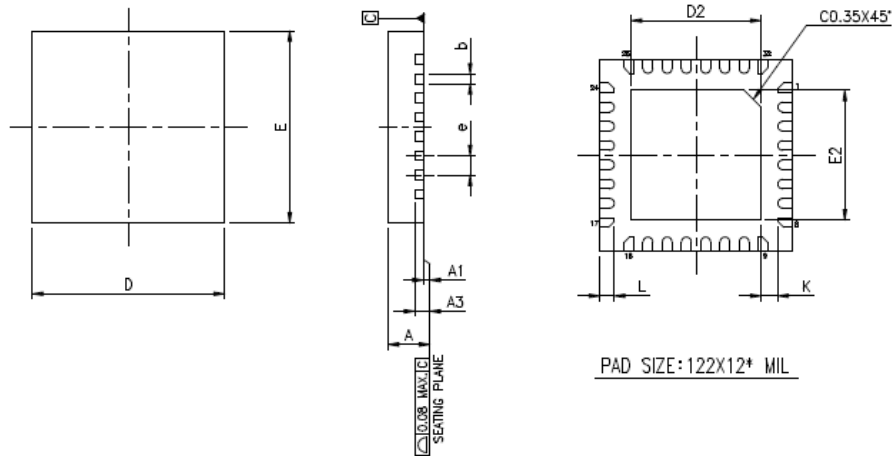
## 16.5 QFN 28 PIN



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	0.003	0.030	0.031	0.07	0.75	0.80
A1	0.000	0.001	0.002	0.00	0.02	0.05
A3	0.008 REF.			0.20 REF.		
b	0.006	0.008	0.010	0.15	0.20	0.25
D	0.16 BSC			4.00 BSC		
E	0.16 BSC			4.00 BSC		
e	0.016 BSC			0.40 BSC		
L	0.014	0.016	0.018	0.35	0.40	0.45
K	0.008	-	-	0.20	-	-

PAD SIZE	D2 (mm)			E2 (mm)		
	MIN	NOR	MAX	MIN	NOR	MAX
115x115 MIL	2.50	2.60	2.65	2.50	2.60	2.65

## 16.6 QFN 33 PIN



**NOTES :**

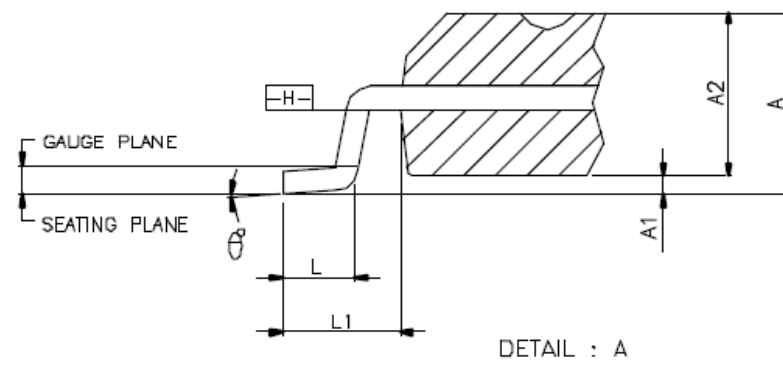
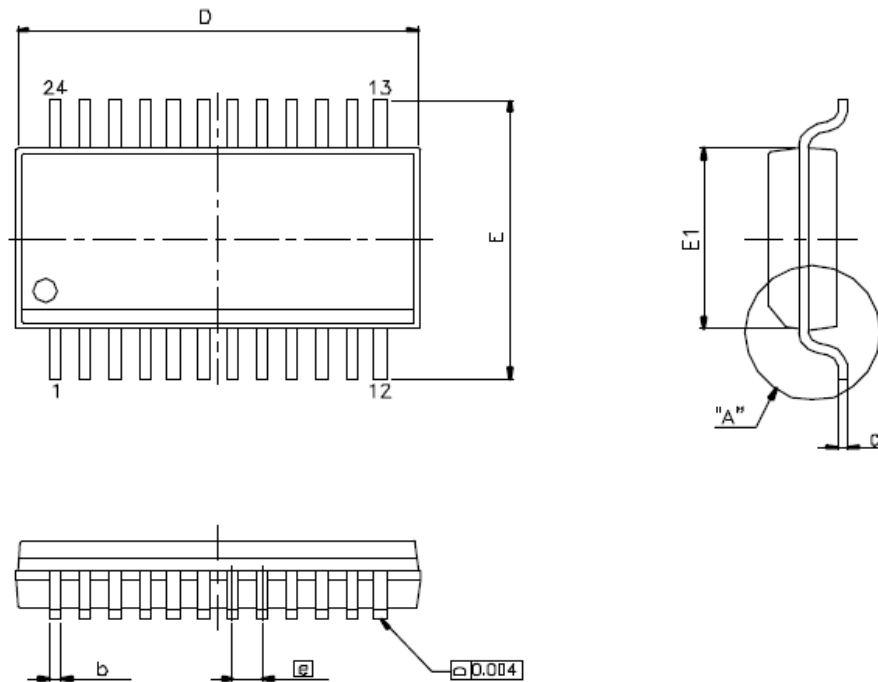
1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSION b APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.15mm AND 0.30mm FROM THE TERMINAL TIP. IF THE TERMINAL HAS THE OPTIONAL RADIUS ON THE OTHER END OF THE TERMINAL, THE DIMENSION b SHOULD NOT BE MEASURED IN THAT RADIUS AREA.
3. BILATERAL COPLANARITY ZONE APPLIES TO THE EXPOSED HEAT SINK SLUG AS WELL AS THE TERMINALS.

SYMBOLS	MIN.	NOM.	MAX.
A	0.70	0.75	0.80
A1	0.00	0.02	0.05
A3	0.20 REF.		
b	0.15	0.20	0.25
D	4.00 BSC		
E	4.00 BSC		
e	0.40 BSC		

PAD SIZE	D1			D2			D3			E1			E2			E3			E4		
	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.
114X11* MIL	—	—	—	2.65	2.70	2.75	—	—	—	—	—	—	2.65	2.70	2.75	—	—	—	—	—	—
122X12* MIL	—	—	—	2.65	2.70	2.75	—	—	—	—	—	—	2.65	2.70	2.75	—	—	—	—	—	—

PAD SIZE	L			L1			K			K1			LEAD FINISH		JEDEC CODE
	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	MIN.	Pure Tin	PPF	
114X11* MIL	0.35	0.40	0.45	0.332	0.382	0.432	0.20	—	—	—	—	—	V	X	N/A
122X12* MIL	0.25	0.30	0.35	—	—	—	0.20	—	—	—	—	—	V	X	N/A

## 16.7 SSOP 24 PIN



SYMBOLS	MIN.	NOM.	MAX.
A	0.053	0.064	0.069
A1	0.004	0.006	0.010
A2	—	—	0.059
D	0.337	0.341	0.344
E	0.228	0.236	0.244
E1	0.150	0.154	0.157
b	0.008	—	0.012
c	0.007	—	0.010
e	0.025 BASIC		
L	0.016	0.025	0.050
L1	0.041 BASIC		
$\theta^\circ$	0°	—	8°

UNIT : INCH

### NOTES:

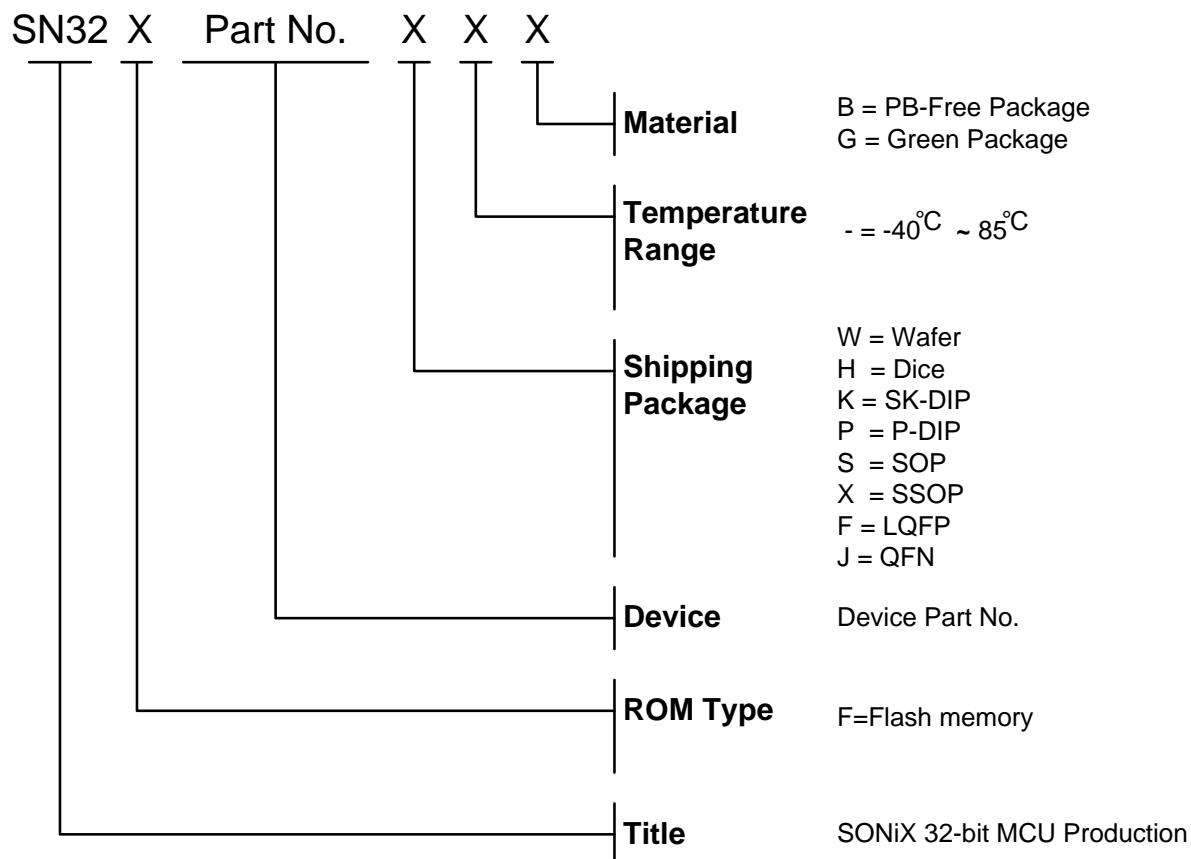
1. JEDEC OUTLINE : MO-137 AE
2. DIMENSION D DOES NOT INCLUDE MOLD PROTRUSIONS OR GATE BURRS. MOLD PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED 0.006" PER SIDE. DIMENSION E1 DOES NOT INCLUDE INTERLEAD MOLD PROTRUSIONS. INTERLEAD MOLD PROTRUSIONS SHALL NOT EXCEED 0.010" PER SIDE.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION/INTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.004" TOTAL IN EXCESS OF b DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR INTRUSION SHALL NOT REDUCE DIMENSION b BY MORE THAN 0.002" AT LEAST.

# 17 芯片正印命名规则

## 17.1 概述

SONiX 32 位单片机产品具有多种型号，本章将给出所有 32 位单片机分类命名规则。

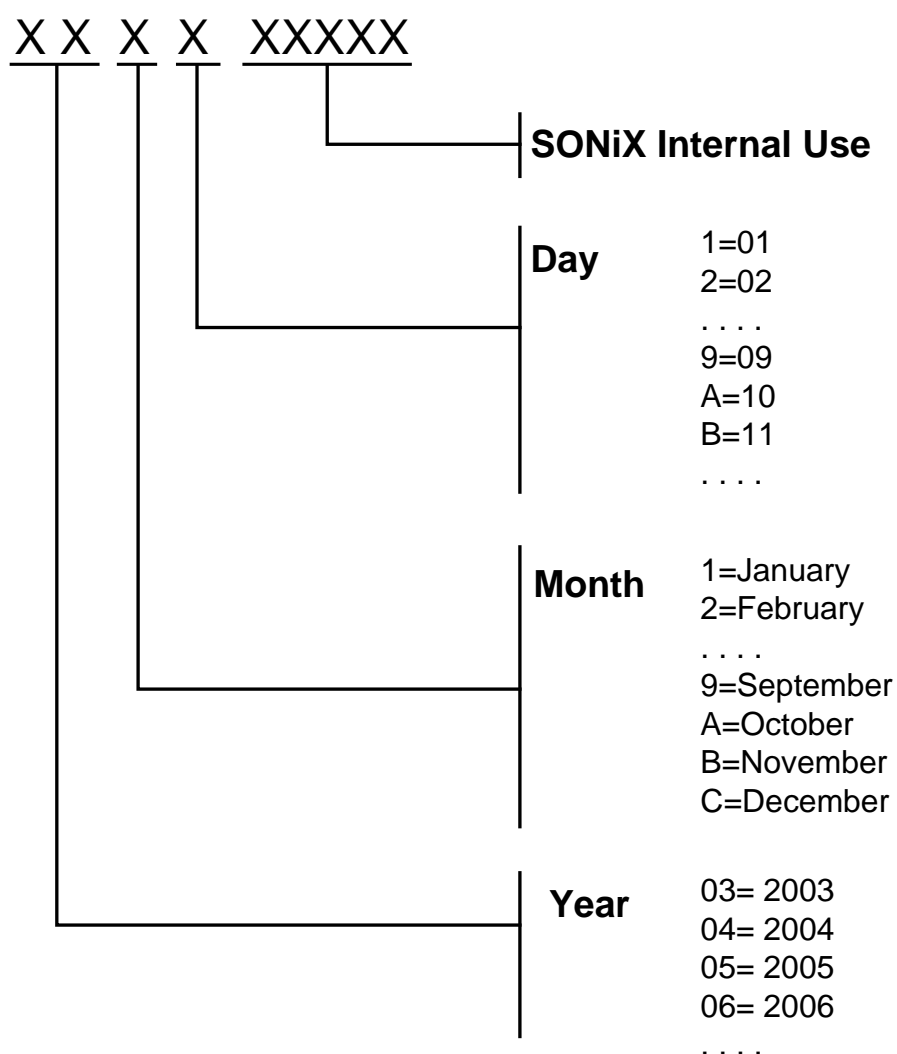
## 17.2 芯片型号说明



## 17.3 命名举例

Name	ROM Type	Device	Package	Temperature	Material
SN32F268FG	Flash memory	268	LQFP	-40℃~85℃	绿色封装
SN32F267JG	Flash memory	268	QFN	-40℃~85℃	绿色封装
SN32F265JG	Flash memory	268	QFN	-40℃~85℃	绿色封装
SN32F2641JG	Flash memory	268	QFN	-40℃~85℃	绿色封装
SN32F264SG	Flash memory	268	SOP	-40℃~85℃	绿色封装
SN32F264XG	Flash memory	268	SSOP	-40℃~85℃	绿色封装
SN32F263XG	Flash memory	268	SSOP	-40℃~85℃	绿色封装
SN32F268W	Flash memory	268	Wafer	-40℃~85℃	-
SN32F268H	Flash memory	268	Dice	-40℃~85℃	-

## 17.4 日期码规则



SONiX 公司保留对以下所有产品在可靠性，功能和设计方面的改进作进一步说明的权利。SONiX 不承担由本手册所涉及的产品或电路的运用和使用所引起的任何责任，SONiX 的产品不是专门设计来应用于外科植入、生命维持和任何 SONiX 产品的故障会对个体造成伤害甚至死亡的领域。如果将 SONiX 的产品应用于上述领域，即使这些是由 SONiX 在产品设计和制造上的疏忽引起的，用户应赔偿所有费用、损失、合理的人身伤害或死亡所直接或间接产生的律师费用，并且用户保证 SONiX 及其雇员、子公司、分支机构和销售商与上述事宜无关。

总公司：

地址：台湾新竹县竹北市台元街 36 号 10 楼之一

电话：886-3-5600-888

传真：886-3-5600-889

台北办事处：

地址：台北市松德路 171 号 15 楼之 2

电话：886-2-2759 1980

传真：886-2-2759 8180

香港办事处：

地址：香港新界沙田炭禾盛街 11 号，中建电讯大厦 26 楼 03 室

电话：852-2723 8086

传真：852-2723 9179

松翰科技（深圳）有限公司

地址：深圳市南山区高新技术产业园南区 T2-B 栋 2 层

电话：86-755-2671 9666

传真：86-755-2671 9786

技术支持：

Sn8fae@SONiX.com.tw